TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

TVORBA SYSTÉMU ROZPOZNÁVÁNÍ ŘEČI PRO ANGLIČTINU

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Lukáš Matějů**
Vedoucí práce: Ing. Petr Červa, Ph.D.

Liberec 2014

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

BUILDING OF A SPEECH RECOGNITION SYSTEM FOR ENGLISH

Diploma thesis

Study programme: N2612 – Electrical Engineering and Informatics
Study branch: 1802T007 – Information Technology

Author: **Bc. Lukáš Matějů**
Supervisor: Ing. Petr Červa, Ph.D.

Liberec 2014

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Lukáš Matějů**
Osobní číslo:	**M12000210**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Tvorba systému rozpoznávání řeči pro angličtinu**
Zadávající katedra:	**Ústav informačních technologií a elektroniky**

Zásady pro vypracování:

1. Seznamte se s problematikou automatického rozpoznávání řeči z pohledu akustického a jazykového modelování a s metodami pro vyhodnocování úspěšnosti rozpoznávání řeči.
2. Proveďte rešerši dostupných zdrojů dat umožňujících vytvořit akustický a jazykový model pro angličtinu (jedná se o akustická a textová data, slovníky a nástroje pro fonetickou transkripci).
3. Jednotlivé zdroje dat sjednoťte a popřípadě vhodně rozšiřte. Následně použijte co největší množství těchto dat, vhodné nástroje (systém HTK nebo Kaldi) a vlastní skripty k vytvoření akustického a jazykového modelu pro angličtinu.
4. Experimentálně vyhodnoťte kvalitu vytvořených modelů na vhodné testovací sadě a proveďte konverzi nejlepších modelů do formátu použitelného v prostředí programu Newton Dictate.

Rozsah grafických prací:	**Dle potřeby dokumentace**
Rozsah pracovní zprávy:	**cca 40 - 50 stran**
Forma zpracování diplomové práce:	**tištěná/elektronická**

Seznam odborné literatury:

[1] **Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall PTR, May 2001.**

[2] **http://kaldi.sourceforge.net/**

[3] **http://htk.eng.cam.ac.uk/**

Vedoucí diplomové práce:	**Ing. Petr Červa, Ph.D.** Ústav informačních technologií a elektroniky
Konzultant diplomové práce:	**Ing. Jiří Málek, Ph.D.** Ústav informačních technologií a elektroniky
Datum zadání diplomové práce:	**12. září 2013**
Termín odevzdání diplomové práce:	**16. května 2014**

prof. Ing. Václav Kopecký, CSc.
děkan

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
vedoucí ústavu

V Liberci dne 12. září 2013

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Na tomto místě bych chtěl poděkovat všem, kteří se jakkoli podíleli na vzniku této práce. Největší dík patří mému vedoucímu, Ing. Petru Červovi, Ph.D., za pomoc a příkladné vedení mé diplomové práce. Také bych chtěl poděkovat Ing. Jiřímu Málkovi, Ph.D. za užitečné rady při práci na jazykovém modelu. Další dík patří Ing. Karlu Blavkovi za pomoc s metodou pevného zarovnání. Závěrem bych chtěl poděkovat všem, kteří mě podporovali a pomohli mi tak práci zdárně dokončit a odevzdat.

Abstrakt

Práce se zabývá tvorbou systému rozpoznávání řeči pro anglický jazyk z hlediska akustického a jazykového modelování. Práce má teoreticko-praktický charakter s částí věnovanou experimentům. Seznámení se základními přístupy k trénování modelů bylo hlavní náplní teoretické části. Cílem praktické části bylo shromáždění akustických, lexikálních a jazykových dat a za pomoci vhodných nástrojů navržení trénovacích skriptů. Cílem experimentální části bylo vyhodnocení natrénovaných modelů na vhodných testovacích sadách a konverze nejlepších modelů do prostředí aplikace Newton Dictate, která je založená na rozpoznávači vyvíjeném na Technické Univerzitě v Liberci.

Teoretická část představuje základní principy trénování jednotlivých modelů. U akustického modelování jsou nejprve prezentovány jednotlivé kroky vedoucí k vytvoření slovníku, následně je čtenář seznámen s postupy směřujícími až k trénování fonémových skrytých markovských modelů, jejichž princip je také vysvětlen. Část teoretické práce o jazykovém modelování je věnována základům n-gramových modelů a různým metodám vyhlazování. Poslední část se zabývá různými metodami vyhodnocování úspěšnosti rozpoznávání řeči.

V praktické části jsou rozebírány zdroje akustických, lexikálních i jazykových dat, jejich kvalita a dostupnost. Na základě akustických dat byla vybrána americká varianta angličtiny jako hlavní jazyk pro tvorbu modelů. U akustického modelování je čtenář seznámen s vytvořenými skripty pro přípravu dat, zisk fonetických přepisů pomocí pevného zarovnání a trénování modelů pomocí toolkitů HTK a Kaldi. Podobně laděná kapitola je dostupná i pro jazykové modelování.

Experimentální část práce je zaměřená na vyhodnocování různých pokusů nad vytvořenými modely pomocí rozpoznávače vyvíjeného na TUL a pomocí Kaldi. Jednotlivé experimenty odrážejí konkrétní úpravy provedené na zdrojových datech modelů za účelem zlepšení úspěšnosti rozpoznávání. Zároveň experimenty slouží jako porovnání kvality výše zmíněných rozpoznávačů. Úspěšnost rozpoznávání založená na nejlepších modelech přesahuje 65 % u obou zmíněných rozpoznávačů. Zvýšení této hodnoty by bylo možné dalším rozšiřováním zdrojových dat.

Výsledkem práce jsou různé pomocné skripty, případně speciální aplikace, pro získání a předzpracování akustických, lexikálních a jazykových dat. Dalším z výstupů jsou trénovací skripty pro vytváření modelů. Posledním neméně důležitým výstupem jsou nejlépe vyhodnocené modely převedené do formátu podporovaného aplikací Newton Dictate.

Klíčová slova:

akustické modelování, HTK, jazykové modelování, Kaldi, rozpoznávání řeči

Abstract

This paper is dedicated to building a speech recognition system based on acoustic and language modelling. This work is theoretical-practical with experimental part. The main concern of theoretical part is introduction of basic approaches to model training. Acoustic, lexical and text data collection was an important part of practical section of this work. Another practical goal was usage of these data and appropriate tools to design training scripts. Evaluation of trained models using different testing data was a major goal of experimental part. The best created models were converted into file format supported by Newton Dictate application, which is based on recognition system developed at Technical University of Liberec.

Theoretical section of this paper presents basic principles of model training. Firstly the individual steps leading to creation of dictionary are described followed by process heading towards to training of phoneme Hidden Markov Models. Its basics are also explained. Another section is dedicated to basics of n-gram models and different techniques of smoothing. The last bit is focused on evaluation of accuracy of speech recognition systems.

The source, quality and availability of acoustic, lexical and text data are analysed in practical part of this paper. Due to the nature of these data American English was chosen as the main language for all models. The reader is presented with scripts designed for acoustic data preparation, phonetic label gathering based on forced alignment and model training using toolkits HTK and Kaldi. Similar chapter is also available for language modelling.

The experimental part of this paper is focused on evaluation of different experiments based on created models using speech recognizer developed at TUL and recognizer presented in toolkit Kaldi. All of the experiments reflects individual changes made to models and source data in order to achieve higher recognition accuracy. These experiments could be also used as a comparison of recognizers mentioned above. The accuracy of recognition based on the best created models is over 65 % for both recognizers. Higher numbers of this value could be achieved by adding additional source data.

The results of this work are scripts and special applications for gathering and preprocessing of acoustic, lexical and text data. The training scripts for creating models are another output of this work. Last but not least, the best experimentally chosen acoustic and language models converted into file format supported by Newton Dictate are output of this paper.

Keywords:
Acoustic Modelling, HTK, Kaldi, Language Modelling, Speech Recognition

Obsah

Seznam obrázků

Seznam grafů

Seznam tabulek

Seznam symbolů, zkratek a termínů

AM	Acoustic Model, akustický model
ARPA	formát pro ukládání n-gramového jazykového modelu
ASCII	American Standard Cod efor Information Interchange, standardizovaná znaková sada
DFT	Discrete Fourier Transformation, diskrétní Fourierova transformace
DTW	Dynamic Time Warping, algoritmus hledající nejkratší cestu mezi dvěma slovy
G2P	Grapheme-to-Phoneme, převod textové podoby řeči do fonetické
HMM	Hidden Markov Models, skryté markovské modely, nejpoužívanější statistický přístup k tvorbě akustických modelů
HTK	The Hidden Markov Model Toolkit, toolkit pro úlohy zpracování řeči
GMM	Gaussian Mixture Model, model využívající mixtury
IFDT	Inverse Discrete Fourier Transformation, inverzní diskrétní Fourierova transformace
IPA	International Phonetic Alphabet, mezinárodní fonetická abeceda
IRSTLM	IRST Language Modelling Toolkit, toolkit pro tvorbu jazykových modelů
Kaldi	toolkit pro úlohy zpracování řeči
lex	formát slovníků používaný na TUL
LM	Language Model, jazykový model
LPC	Linear Prediction Coefficients, metoda odhadu kepstrálních příznaků
MFCC	Mel-Frequency Cepstral Coefficients, metoda výpočtu kepstrálních příznaků
MITLM	MIT Language Modeling toolkit, toolkit pro tvorbu jazykových modelů
SLM	Stochastic Language Model, stochastický jazykový model
TIMIT	akustický korpus americké angličtiny
TTS	Text to Speech, systém pro převod textu do mluvené podoby
VoxForge	akustický korpus
WAV	Waveform Audio File Format, formát pro ukládání zvuku
XML	Extensible Markup Language, obecný značkovací jazyk

Úvod

Nejpřirozenějším a nejvýznamnějším způsobem komunikace mezi lidmi byla a i v dnešní době je řeč. Umožňuje mluvčímu snadno vyjádřit a zároveň rychle předat ostatním poslouchajícím myšlenku. Díky interakci řečníků je také možné okamžitě reagovat na neporozumění obsahu zásahem do aktuálně probíhajícího dialogu. Do řeči o sobě mluvčí promítá další informace, které následně předává okolí, může se jednat například o nářečí, postoj mluvčího k dané promluvě (ironie, sarkasmus), emoce, případně zdravotní stav (kašel). S rozvojem informačních technologií se přirozeně začíná formovat myšlenka komunikace člověka s počítačem pomocí řeči. Vede to až ke vzniku nového progresivního oboru, zpracování řeči. Ten je založený na dvou základních úlohách – syntéze řeči a analýze řeči.

Při syntéze řeči se mluvčím stává počítač, který předává uměle vygenerovanou promluvu příjemci – člověku. Promluva musí být co nejvíce srozumitelná a měla by se také podobat přirozené řeči lidí. Jedná se o úlohu snazší, protože člověk je tvor schopný přemýšlet a spoustu věcí, včetně kontextu, si domyslet. Úloha opačná, rozpoznávání řeči, patří do kategorie analýzy řeči. V tomto případě je příjemcem promluvy člověka počítač, což je hlavním důvodem složitosti tohoto problému. Počítač totiž není schopný porozumět obsahu přenášené zprávy a ignorovat tak nedostatky systému jako člověk. Navíc hlasový signál je silně závislý na mluvčím (nálada, intonace, ironie, barva, tempo řeči…), jeho vyjadřovacích schopnostech (vady řeči) a také na okolí (šum). Úlohu také komplikuje spojitý charakter řeči, kdy nejsou jasně dány hranice jednotlivých slov. Tato práce se zabývá tvorbou systému rozpoznávání řeči pro angličtinu. Mezi další úlohy analýzy řeči patří například rozpoznání řečníka či identifikace jazyka.

Oborem zpracováním řeči se na Technické Univerzitě v Liberci zabývá Laboratoř počítačového zpracování řeči [17] spadající pod katedru Informačních technologií a elektroniky. Tým prof. Ing. Jana Nouzy, CSc. se věnuje rozpoznávání řeči již od začátku 90. let minulého století. Nejprve se zaměřili na úlohu rozpoznávání izolovaných slov, od kterých se postupně přesunuli ke spojité řeči. Dnes má jejich rozpoznávací systém pro český jazyk úspěšnost kolem 90 % při přepisování rozhlasových záznamů a 95-97% úspěšnost u diktování libovolnou

osobou při velikosti slovníku 350 000 slov. Výsledkem jejich práce jsou programy MyDictate a MyVoice určené a speciálně navržené pro handicapované osoby a software Newton Dictate pro záznam diktované spojité řeči vyvinutý ve spolupráci s firmou Newton Technologies a.s.

Cílem teoretické části diplomové práce je seznámení s principy akustického a jazykového modelování a s metodami vyhodnocování úspěšnosti rozpoznávání. Hlavní cílem praktické části je shromáždění akustických, jazykových a lexikálních dat, vytvoření trénovacích skriptů a následné natrénování akustických a jazykových modelů, tedy základních stavebních kamenů pro konkrétní jazyk, pro spojité rozpoznávání angličtiny v prostředí Newton Dictate.

Tato práce je rozdělena do tří základních bloků, první se zabývá teoretickým základem vytváření systému automatického rozpoznávání řeči pro konkrétní jazyk z pohledu akustického a jazykového modelování. Obsahuje také kapitolu o vyhodnocování kvality rozpoznávání. Druhá část je věnována popisu praktické části této práce, od získávání dat až po vytvoření samotných modelů. V poslední části jsou popsány provedené experimenty a vyhodnoceny jejich výsledky v závislosti na různých testovacích úlohách.

1 Teoretická část

Tato kapitola představuje základy tvorby systému rozpoznávání řeči z hlediska akustického a jazykového. Je rozdělena do tří základních bloků. Nejprve je čtenář seznámen s postupy pro vytváření akustických modelů, následně jsou mu představeny základy modelů jazykových. Poslední část této kapitoly se zabývá vyhodnocováním kvality rozpoznávání spojité řeči.

1.1 Akustický model

Kapitola věnovaná akustickému modelování nejprve představuje základní pojmy nutné pro vytváření akustických modelů. Je představena fonetická abeceda a na ní založený slovník. Další součástí této kapitoly je popis parametrizace signálu a následně jsou popsány skryté markovské modely, které jsou v praxi nejčastěji používány k akustickému modelování.

1.1.1 Fonetická abeceda

Základem každého moderního jazyka jsou dvě části – textová a mluvená podoba. Zatímco ta textová je definována grafémy (tvoří abecedu), tak akustickou lze rozložit na nejmenší stavební jednotky – fonémy. Ty je v počítači potřeba reprezentovat také jako znaky. Fonémy tvoří fonetickou abecedu, která umožňuje zformulovat všechna slova daného jazyka. Fonetická abeceda se používá jak při syntéze řeči (systémy Text to Speech – TTS), tak i při analýze, kde má hned několik využití. Pro vytvoření akustického modelu je potřeba znát fonetické přepisy trénovacích vět, na kterých se model trénuje. Aby mohlo být slovo úspěšně rozpoznáno, musí být obsaženo ve slovníku s odpovídající fonetickou transkripcí. V praxi je důležité správně určit optimální velikost fonetické abecedy. Malé množství fonémů nemusí plně pokrývat celou řeč, velké množství naopak může být až příliš podrobné a některé fonémy tak mohou začít splývat. [24]

Nejznámější fonetickou abecedou je Mezinárodní fonetická abeceda IPA (International Phonetic Alphabet) [10], která byla navržena jazykovědci tak, aby pokryla všechny jazyky. Jedná se tak o celosvětový standard využívaný odborníky. Nevýhodou je nutnost používat speciální symboly a zbytečná složitost pro konkrétní jazyky. To vede ke vzniku poměrně velkého množství lokálních fonetických abeced,

které jsou přesně zaměřené na konkrétní jazyk a jeho akustickou podobu. Pro práci se zvukovou podobou angličtiny v této práci byla použita fonetická abeceda odvozená z pravidel abecedy Arpabet [26] pro rozpoznávání spojité angličtiny (viz Tabulka 1). Ta obsahuje méně fonémů než IPA a každému přiřazuje sekvenci znaků z ASCII tabulky.

Tabulka 1: Fonetická abeceda použitá v této práci [26]

foném	slovo	přepis	foném	slovo	přepis
aa	odd	AA D	L	lee	L IY
ae	at	AE T	M	me	M IY
ah	hut	hh AH t	n	knee	N iy
ao	ought	AO t	ng	ping	p ih NG
aw	cow	k AW	ow	oat	OW t
ay	hide	hh AY d	oy	toy	t OY
b	be	B iy	p	pee	P iy
ch	cheese	CH iy z	r	read	R iy d
d	dee	D iy	s	sea	S iy
dh	thee	DH iy	sh	she	SH iy
eh	Ed	EH d	t	tea	T iy
er	hurt	hh ER t	th	theta	TH ey t ah
ey	ate	EY t	uh	hood	hh UH d
f	fee	F iy	uw	two	t UW
g	green	G r iy n	v	vee	V liy
hh	he	HH iy	w	we	W iy
ih	it	IH t	y	yield	Y ie l d
iy	eat	IY t	z	zee	Z iy
jh	gee	JH iy	zh	seizure	s iy ZH er
k	key	K iy			

Poměrná jednoznačnost a malé množství fonémů (20 – 50 pro většinu jazyků) umožňuje vznik různých specializovaných programů pro převod textové podoby konkrétního jazyka do podoby fonetické (G2P aplikace – Grapheme-to-Phoneme). Ty jsou ale častější pro jazyky, kde se zvuková podoba liší jen málo od té písemné. U angličtiny, kde toto neplatí, se jedná o podstatně složitější úlohu.

V praxi se používají i konkrétnější jednotky, například difony (převážně u syntézy řeči), nebo trifony. Ty přidávají kontext, tedy vztah daného fonému vůči fonému předchozímu a budoucímu. To umožňuje lépe postihnout různé výslovnosti hlásek a pomáhá při rozpoznávání řeči. Nevýhodou je ale jejich velké množství, konkrétně u trifonů se jedná o třetí mocninu počtu fonémů.

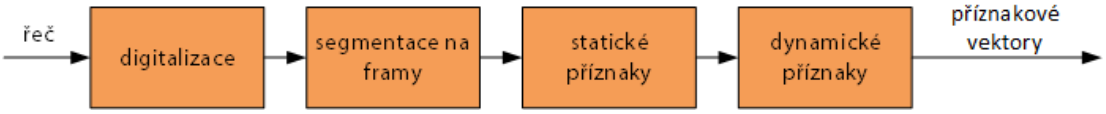
1.1.2 Slovník

Důležitou součástí každého systému rozpoznávání řeči je slovník. Jeho obsah určuje, s jakými slovy bude systém schopný pracovat, tedy rozpoznávat je nebo je moci použít při tvorbě jazykového modelu. Veškeré záznamy ve slovníku mají dvě základní části, samotné slovo a jeho fonetický přepis ve zvolené fonetické abecedě. Dalším parametrem může například být slovní třída přiřazená danému slovu pro snadnější práci při vytváření jazykového modelu. Fonetické varianty slov bývají nejčastěji uváděny jako nový záznam.

Velikost slovníku je vhodné zvolit podle cílové úlohy. Zatímco diktovací úlohy vyžadují velký slovník, ideálně takový, který pokrývá celý jazyk, úlohám ovládání konkrétní aplikace hlasem stačí mnohem menší, sestávající jen z konkrétních příkazů. Dalším důležitým faktorem je tematické zaměření samotné úlohy, které často velikost slovníku značně zmenší. Množství záznamů ve slovníku je také výrazně ovlivněno samotným cílovým jazykem. Ohebné jazyky mohou mít až několikanásobně větší slovní zásobu. Pro češtinu se pohybuje v řádu milionů, zatímco pro angličtinu je dostatečný rozsah kolem 100 000 slov. Slova, která nejsou ve slovníku, nemohou být nikdy správně rozpoznána, obsah slovníku je proto stejně důležitý jako jeho rozsah.

1.1.3 Parametrizace signálu

Cílem parametrizace řečového signálu je získání menšího množství dat, která dobře charakterizují daný signál pro rozpoznávání. Takto získaná data jsou nazývána příznaky. Parametrizace je důležitým krokem každého systému, který značně rozhoduje o úspěchu či neúspěchu rozpoznávání. Obrázek 1 ilustruje sekvenci prováděných kroků s řečovým signálem až po zisk příznakových vektorů pro jednotlivé framy.



Obrázek 1: Proces parametrizace řečového signálu

Při digitalizaci signálu je převáděn analogový signál na svou číslicovou podobu a důležité jsou dva základní parametry – vzorkovací frekvence a kvantizační krok. V praxi se tyto hodnoty nastavují na 16 bitů, respektive 8, případně 16 kHz. Tyto hodnoty jsou dostatečné pro postižení celého spektra lidské řeči.

Takto převedený signál se dále člení na menší jednotky, tzv. framy. Jedná se o krátké úseky (obvykle 10 – 25 ms), jejichž doba trvání by měla být menší než délka trvání nejkratších fonémů. Díky tomu lze signál v daném úseku považovat za stacionární a charakterizovat jej tak menším počtem parametrů. Pro hladký průběh výpočtu parametrů se u framů definuje překryv, kterým zasahují do framů okolních. Rozdělení číslicového řečového signálu na jednotlivé framy se nazývá segmentace.

Dalším krokem je samotná parametrizace, tedy zisk příznakových vektorů pro jednotlivé framy. Zvolené příznaky by měly v ideálním případě potlačovat rysy individuální pro konkrétního mluvčího a zároveň by měly dostatečně odlišit framy nesoucí různé důležité informace. Důležitým faktorem při výběru příznaků je také výpočetní náročnost. S postupem času se přešlo od jednoduchých příznaků (energie, počet průchodů nulou používaných v 60. letech 20. století) přes příznaky spektrální až k příznakům kepstrálním. Ty se poprvé začínají používat kolem roku 1990 a dodnes jsou v praxi rozpoznávání spojité řeči nejpoužívanější.

Kepstrum $c[n]$ lze definovat jako inverzní Fourierovu transformaci logaritmu absolutní hodnoty spektra signálu x:

$$c[n] = \mathrm{DFT}^{-1}\{\log(|\mathrm{DFT}\{x[n]\}|)\} \tag{1}$$

Obrázek 2 zobrazuje základní úkony potřebné k vypočtení kepstra.

Obrázek 2: Výpočet kepstra

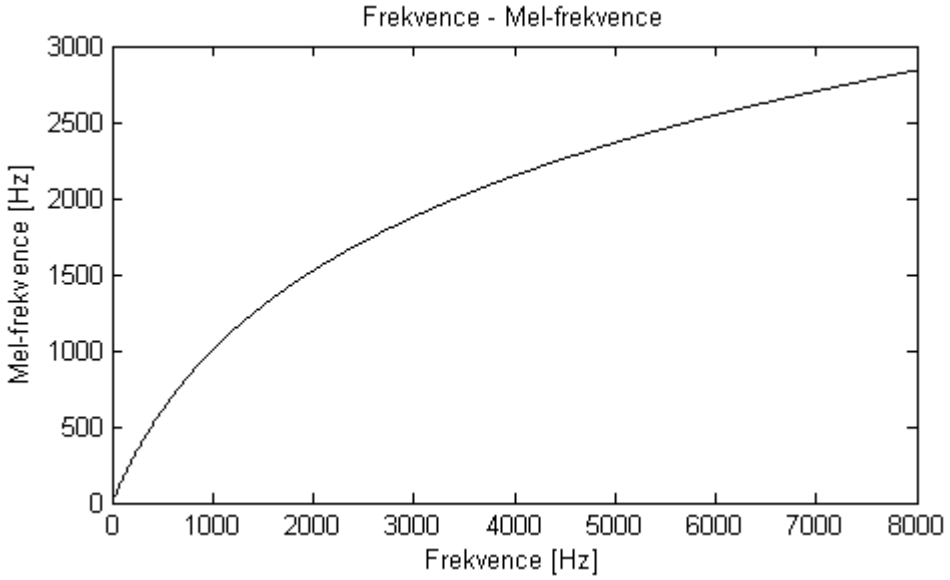
Kepstrální analýza převádí náročnější operaci konvoluce (2) na součet (3), což umožňuje snadnější separaci složek takto vzniklého signálu. Při zpracování řeči lze

řečový signál považovat za výsledek konvoluce periodického (hlasivkového) buzení $s[n]$ s impulsní odezvou hlasového ústrojí $h[n]$. První z těchto informací se transformuje do oblasti vyšších kefrencí, zatímco druhá se soustředí na nízkých kefrencích. Pro zpracování řeči je významná první část, která nese informace o obsahu řeči [28]. Tato část se z kepstra získává liftrací vhodnou okénkovací funkcí, která potlačuje vyšší kefrence.

$$x[n] = s[n] * h[n] \tag{2}$$

$$c[n] = s'[n] + h'[n] \tag{3}$$

V praxi existuje více metod na výpočet kepstrálních příznaků lišících se ve výsledné kvalitě pro rozpoznávání konkrétních úloh. Zejména v 90. letech se využívala metoda LPC odhadující kepstrální koeficienty pomocí lineární predikce v časové oblasti. Tato metoda je méně náročná jak na výpočet, tak na implementaci než dnes rozšířenější MFCC (Mel-Frequency Cepstral Coefficients), ale nedosahuje takových výsledků. MFCC počítá koeficienty přímo podle definice (1), což na současných procesorech již není problém. Zároveň využívá křivky lidského vnímání frekvencí, která není lineární, viz Graf 1. Tato křivka se nazývá Melovská stupnice.



Graf 1: Vztah frekvence a mel-frekvence

Převodní vztah mezi frekvencí f a mel-frekvencí byl stanoven experimentálně:

$$Mel(f) = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \quad (4)$$

Pro normalizaci vypočtených kepstrálních příznaků se někdy používá metoda CMS (Cepstral Mean Substraction), která odečítá průměr příznaků. Zmenšují se tak rozdíly způsobené např. různými mluvčími a hlasitostí. MFCC je dnes používáno v mnoha systémech rozpoznávání spojité řeči, včetně těch vyvíjených na Technické Univerzitě v Liberci.

Příznaky, které byly v této kapitole zatím popsány, počítané pouze na jednotlivých framech, se nazývají statické. Z hlediska rozpoznávání řeči ale může být zajímavý i vztah framu k okolním framům, pak se jedná o tzv. dynamické příznaky. Ty se používají prvního řádu (delta příznaky), případně druhého řádu (delta-delta, akcelerační příznaky). Vyjadřují tedy změnu statických příznaků v čase. Delta příznaky lze spočítat různými způsoby, jedním z nejjednodušších je diference pro konkrétní příznak f_p.

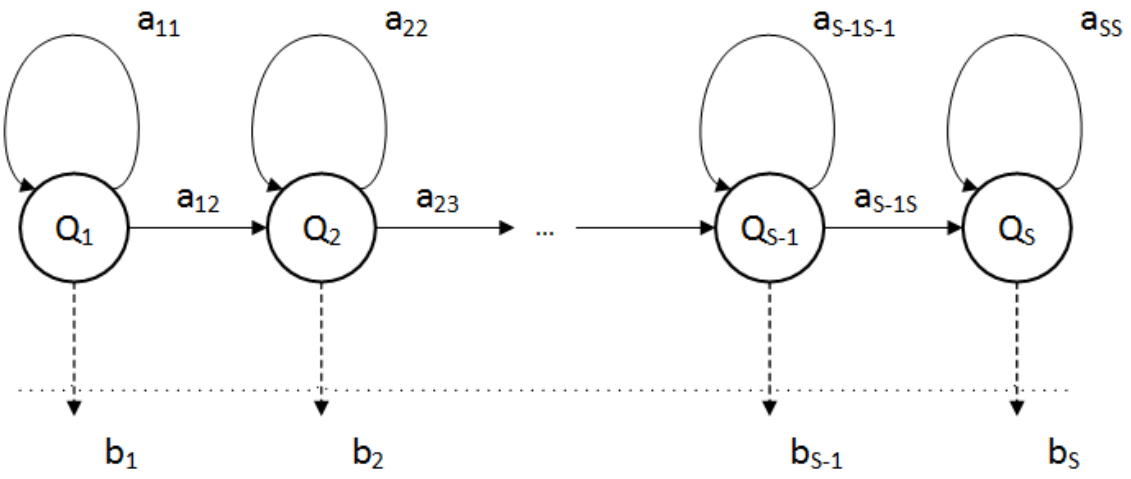
$$\Delta f_p(frame) = f_p(frame + 1) - f_p(frame - 1) \quad (5)$$

Delta-delta příznaky se počítají podle stejného vztahu, vstupem ale jsou delta příznaky. Dynamické příznaky často vedou k lepším výsledkům rozpoznávání [21], proto se v praxi používá kombinace statických i dynamických příznaků.

1.1.4 Skryté markovské modely

Skryté markovské modely (HMM – Hidden Markov Models) jsou jedním z nejvýznamnějších zástupců parametrických modelů, které se používají v oblasti rozpoznávání řeči již od poloviny devadesátých let 20. století [24]. HMM jsou založené na předpokladu, že řeč je v určitých okamžicích (okolních framech) stacionární a v těchto okamžicích se její parametry mění jen málo. To umožňuje přechod od referenčních vzorů používaných u metody DTW k abstraktním modelům, které nahrazují framy menším počtem stavů. Pro všechny framy přiřazené konkrétnímu stavu lze určit statistické rozložení hodnot příznakových vektorů. Nejčastěji se jedná o normální (Gaussovo) rozložení – střední hodnota a rozptyl. Lze také statisticky zjistit, kolik framů obvykle přísluší stavu a určit tak

pravděpodobnost setrvání v daném stavu [8]. Obrázek 3 zobrazuje typickou strukturu modelu.



Obrázek 3: HMM – levo-pravý model

Stavy automatu Q_s jsou uspořádány lineárně, jedná se o takzvaný levo-pravý model. Každý stav má definovanou pravděpodobnost a_{ss} setrvání v daném stavu a pravděpodobnost a_{ss+1} přechodu do stavu dalšího. Pravděpodobnost setrvání ve stavu a pravděpodobnost přechodu do následujícího stavu jsou komplementární jevy, součet jejich pravděpodobností je tedy roven jedné:

$$a_{ss} + a_{ss+1} = 1 \tag{6}$$

Každý stav je také reprezentován pravděpodobnostní výstupní funkcí b_s s normálním rozložením určenou hodnotami rozptylu σ_s^2 a střední hodnoty μ_s. Funkce vyjadřuje míru pravděpodobnosti, že popsaný frame přísluší stavu s. Pro jednorozměrný příznakový vektor x má následující podobu:

$$b_s(x) = \frac{1}{\sqrt{2\pi} * \sigma_s} \exp[-\frac{(x-\mu_s)^2}{2\sigma_s^2}] \tag{7}$$

V praxi se pracuje s vícerozměrnými příznakovými vektory $\boldsymbol{x}$. Pro pravděpodobnostní výstupní funkci b_s platí následující odvozený vztah pro vícerozměrné normální rozložení:

$$b_s(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^P * det\Sigma_s}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\overline{\boldsymbol{x}}_s)^T \sum_s^{-1} (\boldsymbol{x}-\overline{\boldsymbol{x}}_s)\right], \text{kde} \tag{8}$$

Σ_s kovarianční matice určená z hodnot vektorů přiřazených během trénování k danému stavu

$\overline{\boldsymbol{x}}_{\boldsymbol{s}}$ vektor středních hodnot

Pro velké množství trénovacích dat se využívá i vícemodálního normálního rozložení (GMM). Jednotlivé složky jsou označovány jako mixtury. Pro každou mixturu je třeba určit střední hodnotu, rozptyl a váhový koeficient označený *c*. Pravděpodobnostní funkce vypadá následovně:

$$b_s(x) = \sum_{m=1}^{M} c_{sm} \frac{1}{\sqrt{(2\pi)^P \times det\Sigma_{sm}}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \overline{\boldsymbol{x}}_{\boldsymbol{sm}})^T \sum_{sm}^{-1} (\boldsymbol{x} - \overline{\boldsymbol{x}}_{\boldsymbol{sm}})\right], \text{kde} \quad (9)$$

Σ_{sm} kovarianční matice určená z hodnot vektorů přiřazených během trénování k danému stavu

$\overline{\boldsymbol{x}}_{\boldsymbol{sm}}$ vektor středních hodnot

Jednotlivá slova u slovních skrytých markovských modelů jsou reprezentována jedním modelem a počet stavů u různých slov bývá stejný. Slovní HMM jsou vhodné pro aplikace s menším slovníkem, protože pro každé slovo je potřeba nahrát několik záznamů. To je pro aplikace s velkými slovníky značně nepraktické. Pracuje se tedy místo s modely slov s modely menších jednotek, ze kterých se slova skládají, v dnešní době nejčastěji s fonémy – monofony nebo trifony. Pro vytvoření kvalitních modelů je potřeba velké množství foneticky bohatých vět daného jazyka s mnoha výskyty všech fonémů v různém kontextu. Jednotlivé fonémy jsou značně akusticky variabilní, proto má výstupní pravděpodobnostní funkce často tvar vícemodálního normálního rozložení (9).

Modely fonémů jsou pro HMM nejčastěji třístavové, kde první stav odpovídá přechodu z předchozího fonému, druhý je samotné jádro fonému a poslední reprezentuje přechod do následujícího fonému. Z takto vytvořených modelů je možné poskládat model pro libovolné slovo slovníku jednoduchým zřetězením jednotlivých fonémových modelů.

1.1.5 Trénování skrytých markovských modelů

Slovní i fonémové modely jsou trénovány podle stejného principu, při trénování jsou hledány parametry *a* (pravděpodobnost přechodu do následujícího stavu

a pravděpodobnost setrvání ve stavu) a b (pravděpodobnostní výstupní funkce) ke všem modelům daného slovníku. Existuje několik základních přístupů k trénování.

Přiřazení framů ke stavům známé

Pokud je přiřazení framů ke stavům známé, lze využít klasických vztahů pro výpočet střední hodnoty a rozptylu. Vztahy pro jednorozměrný příznakový vektor x jsou:

$$\mu_s = \frac{1}{N_s}\sum_{n=1}^{N_s} x_n \qquad \sigma_s^2 = \frac{1}{N_s}\sum_{n=1}^{N_s}(x_n - \mu_s)^2 \text{, kde} \tag{10}$$

N_s počet framů přiřazených ke stavu s

Pro vícerozměrný vektor $\boldsymbol{x}$ platí následující:

$$\boldsymbol{\mu_s} = \frac{1}{N_s}\sum_{n=1}^{N_s} \boldsymbol{x_n} \qquad \sum_{\boldsymbol{s}} = \frac{1}{N_s}\sum_{n=1}^{N_s}(\boldsymbol{x_n} - \boldsymbol{\mu_s})(\boldsymbol{x_n} - \boldsymbol{\mu_s})^T \tag{11}$$

Pravděpodobnost přechodu ze stavu Q_s do stavu Q_{s+1} je definována jako:

$$a_{ss+1} = \frac{K}{N_s}\text{, kde} \tag{12}$$

K počet výstupů ze stavu s

Pravděpodobnost setrvání je doplňkem:

$$a_{ss} = 1 - a_{ss+1} \tag{13}$$

Takto získané parametry modelů je možné zpřesňovat pomocí dalších metod.

Přiřazení framů ke stavům neznámé – pevné přiřazení

Pokud není přiřazení framů ke stavům známe, je potřeba tuto informaci získat. Z tohoto důvodu se používá iterační algoritmus založený na pevném zarovnání a Vitterbiho algoritmu. Postup se skládá z následujících kroků:

- inicializační krok,
- přiřazovací krok,
- reestimační krok,
- testování.

V inicializačním kroku jsou framy všech nahrávek daného modelu rovnoměrně přiděleny stavům. Následně jsou určeny prvotní odhady všech parametrů.

Při přiřazovací části se využívá dynamického programování, konkrétně Vitterbiho algoritmu. Při trénování je hledáno pomocí rekurze řešení následujícího vztahu:

$$P(\boldsymbol{X}, \boldsymbol{M}) = \max_{f} \prod_{i=1}^{I} a_{f(i-1)f(i)} b_{f(i)} x(i) \qquad a_{f(0)f(1)} = 1 \tag{14}$$

Pravděpodobnost $\boldsymbol{M}$ pro slovo $\boldsymbol{X}$ je určena jako maximální pravděpodobnost přes všechny přípustné kombinace přiřazení framů a stavů modelu. Vztah je analogický ke vztahu pro DTW, jen se nehledá minimální vzdálenost, ale největší pravděpodobnost.

V následujícím kroku jsou na základě přiřazovacího kroku a vztahů (10), případně (11) určeny nové hodnoty pro střední hodnoty, rozptyly a výstupní přechodové pravděpodobnosti. Pokus se hodnoty liší o více než předem definovaná odchylka ε, algoritmus se vrátí na přiřazovací krok. V opačném případě trénování modelů končí. [24]

Baum-Welchův algoritmus

Vytvořené modely pomocí předchozích dvou metod se často vylepšují aplikací iteračního Baum-Welchova algoritmu. Jedná se o složitější algoritmus, při kterém nejsou framy pevně a výlučně přiřazeny k jednotlivým stavům, ale každý frame se s určitou pravděpodobností může podílet na parametrech všech stavů. Vztahy pro výpočet parametrů tak obsahují navíc ještě okupační pravděpodobnost.

V praxi se pro trénování skrytých markovských modelů používá nejčastěji k inicializaci Baum-Welchova algoritmu takzvaný Flat Start, při němž jsou přes všechny nahrávky určeny hodnoty kovarianční matice – rozptyly. Cílem tohoto kroku je prvotní inicializace parametrů, které budou v dalších krocích zpřesňovány. Následuje iterační část algoritmu, kdy se pomocí Forward-Backward algoritmu zpřesňují jednotlivé HMM parametry. Po ukončení algoritmu jsou výsledkem parametry pro veškeré modely [28]. Tento způsob trénování fonémových skrytých markovských modelů je používán i na Technické Univerzitě v Liberci.

1.2 Jazykový model

Tato kapitola je věnována základům jazykového modelování, je představen nejvýznamnější zástupce pravděpodobnostních modelů. Čtenář je dále seznámen s vybranými metodami pro vyhlazování jazykových modelů.

Jazykový model je v dnešní době nedílnou součástí všech moderních systémů pro rozpoznávání spojité řeči s velkým slovníkem. Slova se stejnou, případně velmi podobnou výslovností, představují pro rozpoznávač značný problém. Rozpoznávač, pracující jen na základě akustiky, nemá v tomto případě prostředky, jak správně vyhodnotit slovo. Tyto prostředky mu poskytuje právě jazykový model, který by měl podle pravidel daného jazyka vybrat vhodné slovo. Pravidla mohou být založena na stochastických i nestochastických metodách. Stochastické jazykové modely pracují na pravděpodobnostním přístupu. Každé posloupnosti slov $W = \{w_1, w_2, \dots, w_n\}$ je přiřazena pravděpodobnost $p(w)$, která je počítána z trénovacích dat (textové korpusy). Nejběžněji používaným pravděpodobnostním modelem je n-gramový jazykový model. [13]

1.2.1 n-gramový model

Stochastické jazykové modely přidělují slovům pravděpodobnosti, ať už pro určení pravděpodobnosti celé věty nebo jen pro odhad následujícího slova v sekvenci slov. Nejjednodušší modely umožňují všem slovům následovat libovolné slovo se stejnou pravděpodobností. Pro jazyk o velikosti slovníku 100 000 by tak byla pravděpodobnost libovolného slova následujícího jiné libovolné slovo 0,000001. Komplexnější modely berou v úvahu četnosti jednotlivých slov v konkrétním jazyce. Libovolná slova mohou stále následovat libovolná slova, ale pravděpodobnost výskytu je dána právě předpočítanými četnostmi. Tento přístup ale znevýhodňuje slova s menší četností i na místech, kde by podle kontextu měla být správně. Pokročilejší modely proto pracují s podmíněnou pravděpodobností slov za předpokladu předchozí sekvence slov. Pravděpodobnost celého řetězce slov $p(w)$ je určována podle řetězového pravidla [19]:

$$p(W) = p(w_1)\, p(w_2|w_1)\, p(w_3|w_1, w_2) \ldots p(w_n|w_1, \ldots, w_{n-1}) =$$

$$= p(w_1, w_2, w_3, \ldots, w_n) = \prod_{i=1}^{n} p(w_i|w_1, w_2, \ldots, w_{i-1}) \qquad (15)$$

Hlavním problémem je, jak určit např. pravděpodobnost $p(w_n|w_1, \ldots, w_{n-1})$. Neexistuje jednoduchý způsob výpočtu pravděpodobnosti slova, kterému předchází dlouhá sekvence slov. Vyžadovalo by to obrovský textový korpus. V praxi se proto tento problém řeší jednoduchou aproximací. Dlouhá sekvence slov je nahrazena n předchozími slovy. Závisí-li slovo pouze na předchozím slovu, hovoří se o tzv. bigramu, $p(w_n|w_{n-1})$. Vztah k předchozím dvěma slovům vyjadřují trigramy, $p(w_n|w_{n-2}, w_{n-1})$. Bigramové, případně trigramové, modely jsou nejpoužívanější v systémech rozpoznávání řeči. Konkrétně se u ohebných jazyků používají bigramy, protože slovní zásoba těchto jazyků je několikanásobně větší než u neohebných jazyků. U těch se někdy mohou využívat i trigramy, opět ale v závislosti na velikosti slovníku. Používání vyšších řádů by bylo prospěšné, ale je silně omezeno nároky na paměť a výkon. V jiných úlohách mohou mít význam i vyšší řády, například při rozpoznání jazyka postaveném na fonémových jazykových modelech. Následuje ukázka pravděpodobnosti pro slovo year v n-gramovém modelu s různým n.

```
Příklad:
věta: She had your dark suit in greasy wash water all year.

Ideální případ:
  p(year | She had your dark suit in greasy wash water all)

trigram:
  p(year | water all)

bigram:
  p(year | all)

unigram
  p(year)
```

Vytvořený model nabývá rozměrů velikosti slovníku o dimenzi n. Například pro bigramový model se slovníkem o velikosti 100 000 je tedy nutné alokovat matici o rozměrech 100 000 × 100 000. Jednotlivé hodnoty n-gramů této matice jsou odhadnuty na základě četností z trénovacího korpusu. Ten musí být dostatečně

velký a v ideálním případě jazykově bohatý. Měl by také být tematicky zaměřený podle konkrétní požadované úlohy a vhodně předpřipravený (preprocessing). Prvky matice jsou naplněny podle následujícího vztahu pro bigramy:

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})}\text{, kde} \tag{16}$$

$C(w_{n-1})$ počet výskytů slova w_{n-1}

$C(w_{n-1}, w_n)$ počet výskytů dvojic slov w_{n-1}, w_n

Analogicky lze odvodit vztah pro trigramy (17), případně pro vyšší řady n-gramů:

$$p(w_n|w_{n-2}, w_{n-1}) = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})}\text{, kde} \tag{17}$$

$C(w_{n-1}, w_n)$ počet výskytů dvojic slov w_{n-1}, w_n

$C(w_{n-2}, w_{n-1}, w_n)$ počet výskytů trojic slov w_{n-2}, w_{n-1}, w_n

Hodnoty matice jsou podle definice menší jak 1. Výsledná matice je většinou řídká. Nulové hodnoty by u ideálního modelu reprezentovaly n-gramy, které se v daném jazyce vůbec nevyskytují. Ve skutečnosti ale mohou zahrnovat i reálné kombinace, které se jenom neobjevily v trénovacích datech. Pokud se ale daná kombinace objeví v testovacím korpusu, nula způsobí její zamítnutí a chybné rozpoznání. Z tohoto důvodu je potřeba matici po vypočítání pravděpodobností ještě vyhladit. Toho se obecně dosahuje snížením pravděpodobností viděných n-gramů a zvýšením pravděpodobnosti u těch nulových. Součet těchto hodnot musí stále v daném bloku dat dávat hodnotu jedna.

1.2.2 Vyhlazování

Pro vyhlazování jazykových modelů existují různé vyhlazovací metody. Tato kapitola představuje jednu základní ukázkovou a tři s praktickým použitím. Witten-Bell Discounting a Kneser-Ney Discounting jsou využívány i v systémech vyvíjených na Technické Univerzitě v Liberci

Add-One Smoothing

Jedna z nejjednodušších metod vyhlazování je založená na prostém přičtení jedničky k četnostem všech n-gramů a až následném vypočítání pravděpodobností. To zajistí absenci jakékoli pravděpodobnostní nuly. Například pro vyhlazený bigramový model se podmíněné pravděpodobnosti spočítají následovně:

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + V}, \text{kde} \tag{18}$$

$C(w_{n-1})$ počet výskytů slova w_{n-1}

$C(w_{n-1}, w_n)$ počet výskytů dvojic slov w_{n-1}, w_n

V počet slov ve slovníku

Vztah pro vyšší n-gramy lze odvodit analogicky. Nevýhodou tohoto algoritmu je zvýhodňování původních n-gramů s malou pravděpodobností a znevýhodňování těch s větší. Z tohoto důvodu se v praxi používá zřídka. Existuje spousta alternativ, která nepřičítá jedničku, ale čísla menší, různě odvozená.

Witten-Bell Discounting

Algoritmus je založen na poměrně jednoduchém předpokladu, který mu ale umožňuje dosahovat mnohem lepších výsledků než předchozí představená metoda. Předpokládá každý n-gram s četností nula jako jev, který ještě nenastal. Jakmile se tak stane, bude to jeho první výskyt. Z tohoto důvodu může být pravděpodobnost neviděného n-gramu modelována podle pravděpodobnosti prvního spatření n-gramu [13]. Pravděpodobnost prvního spatření n-gramu se spočítá na trénovacím korpusu. Pro neviděné bigramy lze nadefinovat následující vztah:

$$p(w_n|w_{n-1}) = \frac{T(w_{n-1})}{Z(w_{n-1})(C(w_{n-1}) + T(w_{n-1}))}, \text{kde} \tag{19}$$

$C(w_{n-1})$ počet výskytů slova w_{n-1}

$T(w_{n-1})$ počet všech různých dvojic slov, jejichž první slovo je w_{n-1}

$Z(w_{n-1})$ počet dvojic slov, které se neobjevily v datech a jejichž předchůdce je w_{n-1}

Pravděpodobnosti přičtené k n-gramům s nulovou četností je následně třeba odebrat z nenulových n-gramů, aby bylo zachováno, že pravděpodobnost daného bloku je rovna jedné. Opět je uveden vztah pro bigramy:

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1}) + T(w_{n-1})}, \text{kde} \tag{20}$$

$C(w_{n-1})$ počet výskytů slova w_{n-1}

$C(w_{n-1}, w_n)$ počet výskytů dvojic slov w_{n-1}, w_n

$T(w_{n-1})$ počet všech různých dvojic slov, jejichž první slovo je w_{n-1}

Algoritmus selhává u dvojic slov w_{n-1}, w_n, pro které je $T(w_{n-1})$ vyšší než polovina všech druhů slov ve slovníku. Výsledné hodnoty jsou zavádějící. V praxi tento problém nastává málokdy, protože matice bývají řídké. Dále nejsou řešeny případy, kdy se slova w_{n-1} a w_n v trénovacím korpusu neobjeví. Tyto případy je nutné řešit jinou vyhlazovací metodou [24]. Algoritmus Witten-Bell Discounting je implementován například v toolkitu SRILM (kapitola 2.2.3) a je částečně využíván i pro rozpoznávač vyvíjený na Technické Univerzitě v Liberci.

Kneser-Ney Discounting

Další pokročilou metodou vyhlazování je Kneser-Ney Discounting [9]. Tato metoda je přítomná např. v toolkitu SRILM. Je založená na vyhlazovacím principu Absolute Discounting Interpolation, který dále rozšiřuje. Základem je interpolace vyšších a nižších řádů modelů. U vyšších řádů dochází k odečtení hodnoty d od nenulových n-gramů. Vztah pro výpočet pravděpodobnosti vyhlazeného bigramu metodou Absolute Discounting Interpolation je:

$$p(w_n|w_{n-1}) = \frac{\max(C(w_{n-1}, w_n) - d)}{C(w_{n-1})} + \lambda(w_{n-1})p(w), \text{kde} \tag{21}$$

$C(w_{n-1}, w_n)$ počet výskytů dvojic slov w_{n-1}, w_n

$C(w_{n-1})$ počet výskytů slova w_{n-1}

$\lambda(w_{n-1})$ interpolační váha

d odečítaná konstanta

$p(w)$ pravděpodobnost unigramu

Výsledná pravděpodobnost je složena ze dvou částí, první je snížená bigramová pravděpodobnost, hodnoty d mohou být určeny například hodnotami získanými z metody Good-Turing. K této části se přičítá vážená pravděpodobnost unigramu. Hlavní myšlenkou Kneser-Ney Discounting je nahrazení této pravděpodobnosti unigramu něčím více vypovídajícím, konkrétně:

$$p_{continuation}(w) = \frac{|\{w_{n-1}: C(w_{n-1}, w) > 0\}|}{|\{(w_{m-1}, w_m): C(w_{m-1}, w_m) > 0)\}|} \tag{22}$$

V čitateli tohoto vztahu je počítáno, kolik různých slov w_{i-1} se vyskytlo před daným slovem w. Tato hodnota je normalizována počtem všech unikátních bigramů. S využitím tohoto vztahu je možné definovat následující vztah pro Kneser-Ney Discounting:

$$p(w_n|w_{n-1}) = \frac{max(C(w_{n-1}, w_n) - d)}{C(w_{n-1})} + \lambda(w_{n-1}) p_{continuation}(w) \tag{23}$$

Jedinou zbývající neznámou zůstává interpolační váha $\lambda(w_{n-1})$, ta je definována takto:

$$\lambda(w_{n-1}) = \frac{d}{C(w_{n-1})} |\{w: C(w_{n-1}, w) > 0\}| \tag{24}$$

Konstanta d je normalizována počtem výskytů slova w_{n-1} a následně násobena členem reprezentujícím počet různých slov, které mohou následovat za w_{n-1}.

Kneser-Ney Discounting je rozšířením Absolute Discounting Interpolation, nahrazuje pravděpodobnosti unigramů přesnější pravděpodobností $p_{continuation}$. Je samozřejmě možné ho odvodit i pro vyšší řády n-gramů. Tato metoda společně s Witten-Bell Discounting je používána na vyhlazování jazykových modelů i na Technické Univerzitě v Liberci.

Linear Interpolation Smoothing

Metoda Linear Interpolation Smoothing využívá pro odhad podmíněných pravděpodobností n-gramů nižších řádů všech n-gramů. Vzorce odvozené pro bigramy, respektive trigramy vypadají následovně:

$$\hat{p}(w_n|w_{n-1},w_{n-2}) = \lambda_3 p(w_n|w_{n-1},w_{n-2}) + \lambda_2 p(w_n|w_{n-1}) + \\ + \lambda_1 p(w_n) + \frac{\lambda_0}{V} \tag{25}$$

$$\hat{p}(w_n|w_{n-1}) = \lambda_2 p(w_n|w_{n-1}) + \lambda_1 p(w_n) + \frac{\lambda_0}{V}, \text{kde} \tag{26}$$

λ_i lineární koeficient

V počet slov ve slovníku

Vektor λ je stanoven z vedlejších dat (HELDOUT), tato data musí být odlišná od trénovacích, jinak by nejvyšší koeficient byl roven jedné a ostatní nule. Pro výpočet konkrétních hodnot se používá iterační algoritmus Expectation – Maximization (EM). Podmíněné pravděpodobnosti jsou napočítány na trénovacím korpusu. Algoritmus Linear Interpolation Smoothing je také k dispozici v toolkitu SRILM.

1.3 Vyhodnocování

Vyhodnocování kvality rozpoznávání je poměrně jednoduché u izolovaných slov, kde výpočet úspěšnosti je poměr počtu správně rozpoznaných slov ke všem slovům dané promluvy. Úloha se ale komplikuje u spojité řeči, kde se nemusí shodovat počet slov v referenčním přepisu a rozpoznaném textu. V následujícím bloku lze vidět příklad výstupu rozpoznávače i se znaky označujícími situace, které mohou v praxi nastat.

```
Řečeno:      a chosen few will become generals.
Rozpoznáno:    chosen few will become general for.
             D    H    H    H     H      S     I
```

Konkrétně se jedná o následující:

- H – úspěšně rozpoznané slovo (hit),
- S – nesprávně rozpoznané slovo (substitution),
- I – vložené slovo (insertion),
- D – smazané slovo (deletion),
- N – počet všech slov.

Pomocí takto označkovaných výsledků lze nadefinovat následující vztahy pro určení podobnosti referenčního přepisu a rozpoznaného textu [5]:

$$Correctnes = \frac{H}{N} \cdot 100\ [\%] \tag{27}$$

$$Accuracy = \frac{H - I}{N} \cdot 100\ [\%] \tag{28}$$

Úspěšně rozpoznaná slova jsou definována následovně:

$$H = N - D - S \tag{29}$$

Obě míry jsou podobné, liší se jen ignorancí chyb typu inzerce u *Correctnes*. Z tohoto důvodu je ale *Accuracy* považována za přesnější a objektivnější míru a většina prací prezentuje své výsledky právě pomocí ní. Často je také používána míra určující chybovost rozpoznávání – *WER* (Word Error Rate). Ta je definována podle následujícího vztahu a je snadno odvoditelná z *Accuracy*:

$$WER = \frac{D + S + I}{N} \cdot 100 = \ 100 - Accuracy\ [\%] \tag{30}$$

Posledním problémem zůstává, jak správně určit hodnoty pro substituci, inzerci a deleci. Na to lze použít dynamické programování, konkrétně metodu minimální editační chyby (MEE), která pracuje na podobném principu jako DTW. Hledá nejkratší možnou cestu mezi referenčním a rozpoznaným řetězcem s nejmenším počtem oprav. Na určení vzdálenosti se používá Levenshteinova vzdálenost [6], která určuje podobnost dvou řetězců (referenčního a rozpoznaného) a je definována jako počet substitucí, inzercí a delecí potřebných pro stanovení rovnosti těchto řetězců. Před aplikací samotné metody je nutné ručně stanovit váhu daných chyb.

2 Tvorba systému pro rozpoznávání spojité angličtiny

V této kapitole jsou popsány veškeré kroky, které vedly k úspěšnému vytvoření akustických a jazykových modelů pro americkou angličtinu. Modely byly primárně vytvářeny pro rozpoznávací systém vyvíjený na Technické Univerzitě v Liberci. Pro porovnání výsledků experimentů byly také vytvořeny modely pro rozpoznávač obsažený v toolkitu Kaldi. Většina úprav představených v následujících kapitolách tak byla provedena pro oba rozpoznávací systémy.

2.1 Výběr jazyka

V zadání práce není přesně specifikováno, pro jakou variantu anglického jazyka by měly být modely vytvářeny. Z důvodu dostupnosti většího množství anotovaných akustických dat pro americkou angličtinu byla zvolena právě tato varianta a celá práce pro ni byla uzpůsobena.

2.2 Použité technologie

V této části jsou představeny nejdůležitější volně dostupné toolkity použité při tvorbě akustických a jazykových modelů. Některé z nich byly použity i pro úlohy samotného rozpoznávání a následného vyhodnocování experimentů.

2.2.1 Hidden Markov Model Toolkit

Hidden Markov Model Toolkit (HTK) [7] je toolkit pro trénování a manipulaci se skrytými markovskými modely. Jeho hlavním využitím je výzkum v oblasti rozpoznávání spojité řeči, ale lze jej snadno použít i na úlohy příbuzné, např. na rozpoznávání mluvčích. Toolkit HTK tak poskytuje různé nástroje pro trénování akustických modelů, testování a vyhodnocování samotných experimentů. Součástí toolkitu je také rozsáhlý manuál plně popisující všechny možnosti HTK i s příklady [28]. HTK je napsané v programovacím jazyce C a je dostupné jak pro Windows, tak pro operační systémy založené na Unixu. Po bezplatné registraci na oficiálních stránkách dostane uživatel přístup jak k samotnému toolkitu, tak i k manuálu s příklady.

Toolkit byl prvotně vyvinut na univerzitě v Cambridge v roce 1989, kde byl postupně používán pro vytváření systémů rozpoznávání řeči s velkými slovníky.

V současné době jsou k dispozici zdarma zdrojové kódy k dalšímu rozšiřování, ale poslední oficiální verze HTK vyšla v roce 2009. HTK je i dnes oblíbeným nástrojem pro výzkum v oblasti zpracování řeči a je na něm založena celá řada prací včetně této.

2.2.2 Kaldi

Toolkit pojmenovaný po etiopském pastýři koz, který údajně objevil kávovník, je primárně navržený pro výzkum rozpoznávání řeči. Kaldi [15] je napsané v programovacím jazyce C++ a je pod licencí Apache License v2.0, která umožňuje modifikaci zdrojových kódů. Toolkit má podobnou základní funkcionalitu jako HTK a jeho součástí je poměrně rozsáhlý manuál. Součástí Kaldi jsou tzv. recepty pro nejznámější komerční i nekomerční akustické korpusy, které umožňují vytvoření akustických a jazykových (využívá jiných toolkitů, např. SRILM) modelů a následné vyhodnocení experimentů na testovacích datech. Jedná se o dobrý základ pro vytváření receptů vlastních. Kaldi je možné využít i pro jiné úlohy z oblasti zpracování řeči. Toolkit je primárně určen pro unixové operační systémy, verze pro Microsoft Windows jsou vydávány se značným zpožděním. Velikou výhodou tohoto toolkitu je jeho kontinuální vývoj. Kaldi bylo v této práci primárně použito pro ověření správnosti vytvořených modelů a pro porovnání s rozpoznávačem vyvíjeným v Liberci.

2.2.3 SRILM

SRILM (Stanford Research Institute Language Modeling) [25] je toolkit pro vytváření a následné použití statistických jazykových modelů. Hlavní využití nachází v oblasti rozpoznávání řeči, statistickém značkování textu a také v úlohách překladu. Hlavním podporovaným typem modelů jsou n-gramové statistické jazykové modely. První verze toolkitu byla vydána v roce 1995 a jeho vývoj neustále pokračuje. Toolkit SRILM je napsaný kompletně v jazyce C++. Je dostupný pro Windows i Linux a je distribuován pod licencí Research Community License, která umožňuje jeho použití v projektech financovaných jen z výzkumných grantů. K dispozici je také uživatelský manuál popisující hlavní funkcionalitu s příklady.

2.3 Akustický model

Tato kapitola je věnována tvorbě akustických modelů. Nejprve je představena fonetická abeceda popisující americkou angličtinu. Na základě této fonetické abecedy jsou představeny používané slovníky. Dále je část textu také zaměřena na předzpracování akustických dat – konverzi, parametrizaci a filtraci. Filtrace nevhodných akustických dat je prováděna až po sjednocení fonetických přepisů. Pro jejich zisk je představen princip pevného zarovnání. V závěru této kapitoly je část textu věnovaná samotnému trénování fonémových skrytých markovských modelů. Tvorba těchto modelů je popsána z pohledu toolkitů HTK a Kaldi. Na základě prvního jmenovaného jsou trénovány akustické modely i na Technické Univerzitě v Liberci.

2.3.1 Fonetická abeceda

Fonetická abeceda pro angličtinu použitá v této práci je k nahlédnutí v kapitole 1.1.1. Tato sada byla vybrána kvůli slovníku CMU Dictionary [26], který se stal základem pro finální slovník. K této fonetické sadě byly pro účely rozpoznávání přiřazeny znaky reprezentující ticho a různé hluky, konkrétně 5.

Formát `lex` používaný pro slovníky v systémech na Technické Univerzitě v Liberci neumožňuje reprezentaci fonému dvěma znaky. Následující konverze proto převádí původní značení fonémů na nové reprezentované pouze jedním znakem. Zároveň byla tato konverze navržena tak, aby znaková podoba přepisu co nejvíce odpovídala té zvukové. Některé fonémy angličtiny jsou ale mnohem lépe popsatelné alespoň dvěma znaky, není konverze z hlediska čitelnosti pro uživatele vždy ideální. Tabulka 2 obsahuje pravidla pro konverzi mezi formáty HTK a `lex`.

Tabulka 2: Konverze mezi HTK a lex formátem

HTK	Lex	HTK	Lex	HTK	Lex
aa	o	f	f	p	p
ae	e	g	g	r	r
ah	a	hh	h	s	s
ao	Á	ih	i	sh	š
aw	A	iy	í	t	t

HTK	Lex	HTK	Lex	HTK	Lex
ay	á	jh	Č	th	T
b	b	k	k	uh	u
ch	č	l	l	uw	ú
d	d	m	m	v	v
dh	D	n	n	w	w
eh	É	ng	N	y	y
er	E	ow	O	z	z
ey	é	oy	ó	zh	ž

Různé hluky a vady řeči jsou běžnou součástí lidské řeči. Je potřeba, aby na ně rozpoznávač dokázal také adekvátně reagovat. Z tohoto důvodu jsou i hluky reprezentovány v akustických modelech. Podobně důležitou součástí každé promluvy je i správně anotované ticho. Problém ale může být získání akustických dat s anotovanými hluky a tichem. Tabulka 3 zobrazuje přehled hluků a ticha použitých v této práci, značení odpovídá tomu použitému na TUL pro češtinu.

Tabulka 3: Tabulka ticha a hluků

HTK	Lex	popis	HTK	Lex	popis
si	-	ticho	n3	3	dech
n0	0	úder	n4	4	ruch
n1	1	mlasknutí	n5	5	ehm

2.3.2 Slovník

Základem pro použité slovníky byla fonetická abeceda představená v kapitole 1.1.1 a slovníky CMU Dictionary [26] a TIMIT [18]. Druhý ze slovníků používá rozsáhlejší fonetickou abecedu, která byla převedena do standardní 39 znakové sady použité v celé práci (kapitola 2.3.6). Takto upravený slovník TIMIT byl použit pro experimenty prováděné čistě na akustickém korpusu TIMIT. Pro ostatní experimenty byl sestaven slovník větší, který vznikl převážně sjednocením slovníků CMU Dictionary a TIMIT. Dále byla doplněna část slov ze slovníku distribuovaného s databází VoxForge a pro několik podstatných slov byl vygenerován fonetický přepis pomocí nástroje LOGIOS [16] a následně ručně zkontrolován. Jednalo se

o slova, jejichž chybějící transkripce by způsobila vyřazení až několika hodin trénovacích záznamů.

Tabulka 4 demonstruje rozdíly ve velikosti obou slovníků. TIMIT také obsahuje ke každému slovu pouze jednu fonetickou variantu, zatímco složený slovník má pro některá foneticky odlišnější slova variant více. Složený slovník by měl být svým rozsahem dostačující pro popsání anglického jazyka a díky tomu vhodný pro diktovací aplikace.

Tabulka 4: Porovnání slovníků

slovník	unikátní slova	počet slov
TIMIT	6 231	6 231
složený	125 061	134 415

Slovníky pro toolkity HTK a Kaldi mají jednoduchou strukturu. Každé slovo je na novém řádku a jeho fonetická transkripce je od něj oddělena tabulátorem, respektive mezerou. Samotné fonémy jsou od sebe odděleny mezerou. Slova s výslovnostními variantami jsou uvedeny na dalším samostatném řádku. Ukázka záznamů z formátu pro HTK a Kaldi:

```
Jewelry jh uw l r iy
jewels  jh uw ah l z
```

Systémy pro zpracování řeči na TUL pracují se slovníky ve formátu `lex`. Jedná se o klasický XML dokument s kořenovým elementem `nanolexicon`. Podřazené jsou mu nepárové tagy `item`, které mohou mít až tři atributy. Povinnými atributy jsou `ft` a `p`, které obsahují samotné slovo, respektive jeho fonetický přepis bez oddělovačů. Tento styl zápisu neumožňuje reprezentovat jednotlivé fonémy více znaky. Na nutnou konverzi z formátů HTK, případně Kaldi byl napsán jednoduchý skript v programovacím jazyce Perl, který převede kompletní slovník do formátu `lex`. Konverzní tabulka je popsána v kapitole 2.3.1. Poslední a zároveň nepovinný atribut `m` určuje, jak bude dané slovo reprezentováno. Podobně jako v případě formátu pro HTK jsou výslovnostní varianty každá na novém řádku.

```
<?xml version='1.0' encoding='UTF-8'?>
<nanolexicon>
<item ft='zero' p='zirO' m='0#' />
<item ft='zero' p='zírO' m='0#' />
…
</nanolexicon>
```

Tyto základní varianty slovníků jsou následně ještě rozšiřovány o specifické slovníky vzniklé při předzpracování textů jazykového modelu. Rozšíření se nejčastěji týkají např. číslovek, dat a titulů.

2.3.3 Akustická data

Důležitým základem každého akustického modelu pro diktovací systémy pro přepis spojité řeči je velké množství rozmanitých a zároveň kvalitních akustických dat. V ideálním případě alespoň několik desítek až stovek hodin od co největšího množství mluvčích. Získat takováto data není lehkým úkolem, většina akustických korpusů je komerčních a kvalita volně dostupných zdrojů je kolísavá. Pro trénování jsou navíc potřeba fonetické transkripce daných promluv a to zisk různorodých dat ještě více komplikuje. Pro tuto práci byla použita data z komerčního korpusu TIMIT a volně šiřitelné databáze VoxForge. Ta ale nemá foneticky anotované promluvy. Kvalita těchto dat výrazným podílem ovlivňuje úspěšnost rozpoznávání.

TIMIT

Korpus TIMIT [18], prodávaný konsorciem LDC (Linguistic Data Consorcium), poskytuje akustická a fonetická data pro vývoj a testování systémů automatického rozpoznávání řeči. Na nahrávání se podílelo 630 mluvčích, z nichž každý namluvil 10 foneticky různorodých vět. Mluvčí byli vybráni tak, aby pokryli osm nejvýznamnějších dialektů americké angličtiny a aby byla zároveň zahrnuta obě pohlaví. Takto bylo nahráno 5,4 hodin audio dat (16 kHz, 16 bit), která byla následně rozdělena na trénovací a testovací podmnožinu. Ke každé nahrávce jsou k dispozici základní informace o mluvčím (pohlaví, dialekt), textový a fonetický přepis s časovými značkami. Fonetická abeceda se skládá z 61 znaků a je nazvána TIMITBET. V praxi se ale často používá fonetická sada menší a je nutná konverze. Součástí korpusu TIMIT je také slovník obsahující všechna slova z textových přepisů

s fonetickou transkripcí. Promluvy jsou ve většině případů velmi krátké, jen několik vteřin dlouhé. Každá promluva vypadá následovně.

```
txt soubor
  she had your dark suit in greasy wash water all year

phn soubor
  0 9640 h#
  9640 11240 sh
  11240 12783 iy
  …

wrd soubor
  9640 12783 she
  12783 17103 had
  17103 18760 your
  …

wav soubor
```

TIMIT je často používán na různé experimenty rozpoznávání řeči, případně identifikace mluvčích, díky ručně zkontrolovaným fonetickým přepisům [11]. Zároveň je poměrně malý, aby bylo možné pouštět větší množství experimentů v rozumném čase, ale dostatečně velký, aby demonstroval potenciál daného systému.

VoxForge

Cílem projektu VoxForge [3] bylo získání velkého množství audio dat s transkripcí vět, která by bylo možné použít na vytvoření akustických modelů pro volně šiřitelné rozpoznávací systémy (Sphinx, ISIP, Julius). Nahrávky jsou vytvářeny komunitou, pro usnadnění práce je možné nahrávat přímo na stránkách projektu pomocí pluginu. K datům jsou k dispozici základní informace o mluvčím – pohlaví, dialekt. Veškerá tato data jsou pod licencí GPL [4] a jsou volně dostupná ke stažení v různé kvalitě (vzorkovací frekvence, bitová hloubka, formát). VoxForge dnes obsahuje přibližně 100 hodin mluvené angličtiny různých dialektů, v menší míře jsou dostupná i data pro další jazyky. Součástí je také slovník s fonetickou transkripcí vycházející z CMU Dictionary. VoxForge neobsahuje testovací množinu, velmi často je náhodně vybírána podmnožina ze všech dat a ta je následně použita k testovacím účelům. Promluvy jsou podobně jako u korpusu TIMIT velmi krátké.

VoxForge je alternativou ke komerčním akustickým korpusům, ale neobsahuje fonetické přepisy jednotlivých nahrávek. Kvůli obecné otevřenosti nahrávání dat je také různá kvalita samotných dat. Velké množství z nich je značně znehodnoceno šumem nebo jsou nahrány nerodilými mluvčími se silnými přízvuky. Data je proto vhodné před použitím filtrovat.

2.3.4 Konverze akustických dat

Akustická data bylo nutné nejprve sjednotit. Pro trénování pomocí toolkitů HTK i Kaldi jsou používána data ve formátu `LPCM` uložená v kontejneru `WAV`. Všechny nahrávky jsou vzorkovány frekvencí 16 kHz a kvantizační krok je 16 bitů. Stejné údaje se používají při zpracování češtiny na TUL a jedná se o jeden ze standardů v oblasti zpracování řeči. Korpus TIMIT sice obsahuje nahrávky s parametry 16 kHz a 16 bit, ale v kontejneru `WAV` jsou data ve formátu `NIST`. Pro konverzi do `LPCM` byl použit oficiální skript dodávaný s korpusem TIMIT. Databáze VoxForge distribuuje data v různých formátech s různými parametry, mezi nimiž je i požadovaná konfigurace. Ta byla tedy použita, aby nebyla prováděna žádná konverze navíc. Takto sjednocená data je už možné použít na trénování, je ale vhodné je nejdřív profiltrovat.

2.3.5 Parametrizace

Pro parametrizaci byly použity kepstrální příznaky, konkrétně MFCC, kapitola 1.1.3. Pro systémy vyvíjené na Technické Univerzitě v Liberci jsou počítány programem `MelCepParam3916`, jehož vstupem je soubor ve formátu popsaném v předchozí kapitole. Výstupem je pak příznakový vektor uložený ve formátu `MFCC3916`, se kterým pracují další části systému. Pro dávkové zpracování všech audio souborů byl napsán jednoduchý skript, který prochází celou stromovou strukturu akustických dat a počítá postupně kepstrální příznaky pro jednotlivé nahrávky.

Toolkit Kaldi poskytuje pro výpočet kepstrálních příznaků vlastní prostředky, skript `make_mfcc`. Je také počítána varianta MFCC. Kaldi umožňuje dva základní přístupy k počítání MFCC, první je navržený přímo týmem stojícím za Kaldi a v některých částech výpočtu se mírně liší použitím filtrů od implementace

představené v toolkitu HTK. Druhý přístup simuluje výpočet v HTK a jeho výsledkem jsou velmi podobné hodnoty příznakových vektorů. Pro testování v Kaldi byl použit druhý přístup, jehož výsledky jsou podobné těm z programu `MelCepParam3916`. Experimenty by tak v tomto bodě neměly být výrazně ovlivněny.

2.3.6 Konverze fonetické abecedy

Akustická data z korpusu TIMIT jsou anotována za pomoci 61 znakové abecedy TIMITBET. Ta je pro účely rozpoznávání až příliš konkrétní, zároveň pro ni není ani dostupný větší slovník. Z tohoto důvodu bylo potřeba provést konverzi do fonetické abecedy představené v kapitole 1.1.1. Pro konverzi byla použita mírně upravená pravidla používaná v toolkitu Kaldi [15], kde probíhá stejná konverze. Kontrola byla provedena podle manuálu samotného korpusu [18], slovníku [26] a literatury [11], [22], kde byly provedeny podobné konverze. Bezchybnost konverze je klíčová pro úspěšnost rozpoznávání. Jakákoliv zanesená chyba se následně negativně projeví.

Pravidla pro konverzi jsou zobrazena v následující tabulce, foném `dx` je převáděn na foném `d` nebo `t` podle daného slova za pomoci slovníku. Foném `q` je vynechán úplně. Fonémy `bcl`, `dcl`, `gcl`, `kcl`, `pcl` a `tcl` jsou přepsány na ticho. Zároveň jsou sjednoceny všechny značky pro ticho (`sil`, `epi`, `h#`, `pau`) do jednotného ticha `si`. Tabulka 5 obsahuje pravidla pro konverzi, neuvedené fonémy zůstávají ve stejné podobě.

Tabulka 5: Konverze fonetické abecedy

TIMITBET	konverze	TIMITBET	konverze
ao	aa	eng	ng
ax, ax-h	ah	hv	hh
axr	er	ix	ih
dx	d, t (kontext)	ux	uw
el	l	bcl, dcl, gcl, kcl, pcl	si
em	m	tcl, epi, h#, pau, sil	si
en, nx	n	q	

Z důvodu malého rozsahu korpusu TIMIT a absence fonetických přepisů u korpusu VoxForge bylo potřeba získat fonetické přepisy i pro tato data. Na zisk těchto transkripcí byla využita metoda pevného zarovnání popsaná v další kapitole, která umožňuje na hranice slov vkládat ticha, případně hluky.

2.3.7 Pevné zarovnání

Úloha pevné zarovnání (forced alignment) umožňuje získat časově zarovnaná textová data. Rozpoznávač je nucen pracovat se slovy obsaženými pouze v textových přepisech a to v daném pořadí. Vstupem je signál řeči s přepisem. Zároveň je k dispozici slovník, který obsahuje fonetické transkripce pro konkrétní slova přepisů promluv. Na hranice slov je možné vkládat ticha, případně hluky. Výstupem jsou časově značkované textové, případně fonetické přepisy. Pro slova je vybrána nejlepší fonetická varianta. Toolkity HTK i Kaldi jsou schopné provést zarovnání.

Ústav informačních technologií a informatiky používá vlastní nástroj pro pevné zarovnání. Ten umožňuje veškerou výše zmíněnou funkcionalitu včetně vkládání hluků, případně ticha na hranice slov. Vstupními parametry zůstávají samotné nahrávky, jejich přepisy, slovník a také natrénovaný akustický model pomocí skrytých markovských modelů (soubor `hmmdefs`). Ten může být vytvořen například pomocí toolkitu HTK. Tento model byl natrénován na akustických datech z korpusu TIMIT, fonetické přepisy prošly konverzí uvedenou v předchozí kapitole. Trénování akustického modelu bylo provedeno podle kapitoly 2.3.9. Jedním z hlavních úkolů pevného zarovnání byla také anotace hluků. Protože fonetické přepisy korpusu TIMIT neobsahují hluky, byly pro ně použity hodnoty získané trénováním na českých promluvách na TUL. Posledním vstupem je fonetická abeceda. Je potřeba také určit, které hluky a ticha mohou být vkládány na hranice slov. Na přípravu dat pro pevné zarovnání byla vytvořena sada skriptů, která nachystá potřebné soubory. Výstupem pevného zarovnání pro každou promluvu je soubor typu `trsx`, který obsahuje časově zarovnané přepisy. Z pohledu této práce nebyla tato metoda použita ani tak pro zisk časových značek, ale pro zisk správných fonetických přepisů jednotlivých promluv s anotovanými hluky.

Tato operace byla důležitá pro oba korpusy. K akustickým datům TIMIT byly korektně přiřazeny hluky a anotována ticha. Pro korpus VoxForge byly vytvořeny kompletní přepisy na základě ideálních variant ze slovníku a stejně jako u předchozího korpusu byly anotovány hluky a ticha. Pevné zarovnání bylo provedeno dvakrát. Při vyhodnocování prvního průběhu bylo identifikováno velké množství slov, která nebyla obsažena ve slovníku a pro věty je obsahující tak nemohly být správně vygenerovány přepisy. Došlo by tak ke ztrátě až několika hodin akustických dat. Po doplnění záznamů do slovníku (viz kapitola 2.3.2) bylo provedeno druhé pevné zarovnání, s jehož výsledky se dále pracovalo při tvorbě akustických modelů.

Alternativou pro získání fonetických přepisů pro korpus VoxForge jsou různé aplikace pro převod textové podoby do fonetické. Bohužel na rozdíl od češtiny se v angličtině psaná podoba velmi liší od té zvukové a přepis je tak značně obtížnější. Příkladem takové aplikace může být Logios [16] použitý při doplňování slovníku. Nevýhodou je absence možnosti přidat anotace hluků a ticha. Z tohoto důvodu byla pro získání korektních přepisů zvolena metoda pevného zarovnání.

2.3.8 Filtrace akustických dat

Výsledky pevného zarovnání byly také použity k profiltrování akustických dat. Již v předchozích kapitolách bylo zmíněno, že kvalita nahrávek korpusu VoxForge je značně kolísavá, což by mohlo negativně ovlivnit výslednou kvalitu rozpoznávání. Před trénováním bylo potřeba tato data tedy profiltrovat.

Data mohla být vytříděna na základě tří následujících vlastností:

- Pevné zarovnání nedokázalo přiřadit fonetické přepisy.
- Promluva obsahuje velké množství hluků.
- Promluva obsahuje chybějící slova ve slovníku.

Vytvořené skripty pro kontrolu všech tří kritérií procházejí jednotlivé `trsx` soubory vygenerované při pevném zarovnání. Jeden soubor odpovídá jedné promluvě. Jedná se o klasické XML dokumenty, z nichž nejzajímavější je následující část.

```
<p b="570" e="810" f="ší">she </p>
<p b="810" e="1060" f="hed">had </p>
<p b="1060" e="1170" f="yur">your </p>
<p b="1170" e="1300" f="4"></p>
```

Párový tag `p` obsahuje dané slovo promluvy. Atributy `b` a `e` označují začátek a konec slova v promluvě. Nejdůležitější částí z pohledu této práce je atribut `f`, který obsahuje přiřazené fonetické přepisy pro daná slova konkrétní nahrávky.

Skript parsuje `trsx` dokument. Pokud pevné zarovnání nedokázalo přiřadit fonetické transkripce a vyhodnotilo tak nahrávku jako nevhodnou pro trénování, jsou obsahem párových tagů `p` otazníky. Skript vyřadí takto vybrané nahrávky z trénování. Dalším problémem mohou být nahrávky zanesené až přílišným hlukem, šumem okolí. V tomto případě bylo pevné zarovnání sice schopné přiřadit fonetické transkripce, ale tato data stejně nejsou ideální pro trénování. Pro skript bylo potřeba určit práh, při jehož překročení je nahrávka považována za nepoužitelnou. Tento práh byl nastaven experimentálně. Pokud počet hluků překročí počet slov dané promluvy, je nahrávka vyhodnocena jako nevhodná a je vyřazena z trénování. Pokud nahrávka obsahuje slovo, které není ve slovníku, není pro ni vůbec vytvořen soubor `trsx`. Tyto nahrávky jsou samozřejmě také vyřazeny z trénování, jejich počet ale byl omezen na minimum, pouze na nahrávky obsahující málo častá slova se špatně odhadnutelnou výslovností – např. cizí slova.

Filtrace dat se z naprosté většiny týkala korpusu VoxForge, ze kterého bylo vyřazeno přibližně 20 hodin záznamů vyhodnocených jako nevhodné. Z korpusu TIMIT bylo vyfiltrováno jen několik minut, kde pevné zarovnání nebylo schopné správně přiřadit fonetické přepisy. Takto profiltrovaná data byla následně převedena pomocí skriptu zpět do souborů s příponou `phn`. Konkrétně byly pomocí parsování `trsx` dokumentu vybrány fonetické přepisy a následně převedeny do původní abecedy představené v kapitole 1.1.1 rozšířené o hluky a ticho. Tato data už byla následně použita k trénování akustických modelů.

2.3.9 Trénování akustických modelů

Tato kapitola je rozdělena na dvě základní části, první z nich popisuje přípravu vstupních souborů a skripty pro trénování pomocí toolkitu HTK. Druhá část je

zaměřena na toolkit Kaldi a tvorbu akustických modelů pomocí něj. V obou případech se jedná o fonémové skryté markovské modely.

Trénování pomocí HTK

Trénovací skripty používané pro češtinu na Technické univerzitě v Liberci jsou založené na toolkitu HTK. Využívají upravené a vylepšené programy právě z tohoto toolkitu. Pro trénování fonémových skrytých markovských modelů pro angličtinu bylo nutné tyto skripty upravit a vhodně připravit vstupní soubory. Parametry modelů jsou trénovány Baum-Welchovým algoritmem s využitím Flat Startu.

Prvním ze vstupních souborů je soubor `mlf.list`, který obsahuje systémové cesty k jednotlivým parametrizovaným souborům promluv. Každý záznam je uveden na novém řádku. Obsah souboru se liší podle konkrétní úlohy a podle použitých akustických dat. Následuje ukázka jednoduchým skriptem, který prochází stromovou strukturu trénovacích dat a ukládá cesty k parametrizovaným `mfcc3916` souborům, vygenerovaného souboru `mlf.list`.

```
mlf.list
D:/mateju/_other/timit/TRAIN/DR1/FCJF0/SA1.MFCC3916
D:/mateju/_other/timit/TRAIN/DR1/FCJF0/SI1027.MFCC3916
```

Druhým ze vstupů je soubor `mlf.triphones`. Ten obsahuje fonetické transkripce všech promluv v podobě inter-word trifonů. Rozpoznávač vyvíjený na TUL pracuje právě s těmito trifony. Tento typ trifonů dbá na hranice slov. Opakem jsou takzvané crossword trifony, se kterými se lze setkat například u toolkitu Kaldi. Tyto trifony neřeší hranice mezi slovy. Fonémy základní abecedy jsou trénovány jako trifony, hluky a ticho jsou trénovány jako monofony.

```
Příklad:          she       had
přepis :    si    sh iy     hh ae d    si

inter-word trifony          crossword trifony
       si                         si
       sh+iy                      sh+iy
       sh-iy                      sh-iy+hh
       hh-ae                      iy-hh+ae
       hh-ae+d                    hh-ae+d
       ae-d                       ae-d
       si                         si
```

Nejprve byl skriptem vytvořen `mlf` soubor (standard HTK) pro monofony a naplněn přepisy trénovacích dat. Tento soubor postupně obsahuje odkazy na promluvy a následně jejich fonetický přepis.

```
Monophones.mlf
#!MLF!#
"D:/mateju/_other/timit/TRAIN/DR1/FCJF0/SA1.lab"
0 0 si
0 0 sh
0 0 iy
.
```

Z takto vygenerovaného souboru už dokáže HTK příkazem `HLEd` vygenerovat trifonové `mlf`. Jedním z parametrů jsou fonémy, které budou připraveny pouze jako monofony. Pro vytvoření inter-word trifonů je potřeba mezi hranice slov monofonů vložit pomocný symbol jako monofon a ten po převodu následně opět smazat. Takto připravený soubor je použit jako jeden ze vstupů. Obsah tohoto souboru se samozřejmě také liší podle konkrétní trénovací úlohy a použitých dat. Cesty musí odpovídat cestám uloženým v předchozím souboru `mlf.list`. Následuje ukázka `mlf.triphones`.

```
#!MLF!#
"D:/mateju/_other/timit/TIMIT/TRAIN/DR1/FCJF0/SA1.lab"
0 0 si
0 0 sh+iy
0 0 sh-iy
0 0 hh+ae
0 0 hh-ae+d
0 0 ae-d
0 0 si
.
```

Z důvodu velkého množství trifonů se používají takzvané svázané trifony. Trifony jsou podle předem nadefinovaných pravidel shlukovány do skupin reprezentujících podobné trifony. Tato pravidla byla získána přímo z manuálu HTK [28] a následně upravena pro fonetickou abecedu použitou v této práci. Soubor obsahující tato pravidla je nazván `tree.hed` a je nezbytnou součástí skriptů pro vytvoření modelů.

Trénovací skripty jsou rozčleněny do několika základních částí:

- hlavní skript,
- trénování monofonů,
- trénování trifonů,
- svazování a následné trénování svázaných trifonů,
- trénování mixtur,
- volitelně trénování HLDA.

Všechny skripty byly upraveny pro použití na trénování americké angličtiny. Parametry ovlivňující kvalitu výsledného modelu jsou počty iterací u trénování konkrétních fází a počty mixtur. Na Technické Univerzitě v Liberci je nejčastěji používáno 32 mixtur a 8 finálních iterací. I pro většinu provedených experimentů bylo použito toto nastavení. Výstupem je soubor `hmmdefs`, se kterým se dále pracuje. Obsahuje konkrétní hodnoty parametrů pro jednotlivé fonémy. Druhým důležitým souborem je `tiedlist`, který nese informace o provedených svázáních.

Aby bylo možné akustický model použít při rozpoznávání libereckým rozpoznávačem, je potřeba ho převést do formátu `channel`. Tento převod se provádí pomocí programu `hmm2channel`. Vstupem jsou soubory `hmmdefs` a `tiedlist` doplněné o soubor obsahující pravidla pro konverzi fonémů mezi `HTK` a `lex` formátem. Převod do formátu `lex` používaném v celém rozpoznávači je prováděn právě na tomto místě. Posledním vstupem je seznam všech možných kombinací monofonů a trifonů.

Trénování pomocí Kaldi

Součástí Kaldi jsou takzvané recepty pro nejznámější akustické korpusy, které umožňují vytvoření akustických modelů a následné otestování kvality korpusu. Pro vlastní úlohy je nutné tyto recepty samozřejmě značně upravit. Modely jsou trénovány jako crossword trifony, což je hlavním rozdílem oproti trénování pomocí HTK. Ostatní parametry byly zvoleny tak, aby bylo možné výsledky experimentů založených na obou modelech co nejlépe porovnávat.

Prvním ze souborů, které je nutné připravit je soubor nazvaný `text`. Ten obsahuje jednotlivé transkripce přiřazené k textovému řetězci. Konvencí je tento

řetězec definovat ve formátu mluvčí – název promluvy. Tento soubor může obsahovat jak fonetické přepisy, tak i jenom větné. Kaldi si fonetické přepisy odvodí během stavby akustického modelu na základě slovníku, fonetické abecedy a vlastních přepisovacích skriptů. Následuje ukázka ze souboru získaného skriptem, který prochází jednotlivé fonetické přepisy vět a skládá je do jednoho výsledného souboru `text`.

```
FCJF0-SA1 si sh iy hh ae d si
FCJF0-SA2 si d ow n ae s k m iy
```

Druhým důležitým souborem je `wav.scp`. Ten je složen ze dvou částí. První je identifikátor, který odpovídá řetězci z minulého souboru, tedy mluvčí – název promluvy. Druhou částí je cesta k souboru `wav`. Následující soubor byl opět vytvořen skriptem.

```
FCJF0-SA1 D:/mateju/_other/timit/TRAIN/DR1/FCJF0/SA1.wav
FCJF0-SA2 D:/mateju/_other/timit/TRAIN/DR1/FCJF0/SA2.wav
```

Posledním důležitým souborem, který je nutné ručně vytvořit, je `utt2spk`. V něm je definováno, který mluvčí namluvil jakou nahrávku. Formát je promluva mezera mluvčí. Problémem může být neznalost tohoto vztahu, doporučeno je vztah odhadnout, v krajním případě každou promluvu přiřadit unikátnímu mluvčímu. Rozdělení mluvčích bylo pro tuto práci ale známo. Pro trénování v HTK toto rozdělení použito nebylo.

```
SA1 FCJF0
SA2 FCJF0
SX1 FCJF1
```

Pokud je známé pohlaví mluvčího, je vhodné pro lepší výsledky rozpoznávání také vytvořit soubor `spk2gender`. Formát je mluvčí – pohlaví. Pro tuto práci tento soubor vytvářen nebyl.

```
FCJF0 m
FCJF1 f
```

Ostatní potřebné soubory jsou generovány pomocí podpůrných skriptů obsažených přímo v toolkitu Kaldi. S takto připravenými soubory je možné začít trénování akustických modelů. Opět se jedná o skryté markovské modely, i když

toolkit podporuje například i modely založené na neuronových sítích. Byl vytvořen recept, který provádí následující kroky pro vytvoření modelů pro angličtinu:

- parametrizace nahrávek,
- trénování monofonů,
- trénování trifonů,
- trénování svázaných trifonů,
- volitelně trénování LDA,
- volitelně další metody.

Výstupem je akustický model použitelný pro rozpoznávání toolkitem Kaldi. Výsledky rozpoznávání jsou porovnávány s modely vytvořenými pomocí HTK v kapitole 3 věnované experimentům.

2.4 Jazykový model

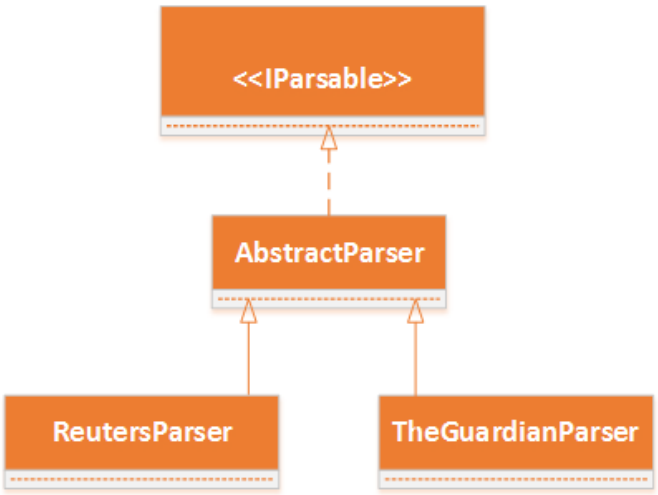
Tato kapitola je zaměřená na praktickou tvorbu jazykového modelu pro systém rozpoznávání spojité řeči pro americkou angličtinu. Popisuje vhodný výběr dat, jejich získání a následné předzpracování. Poslední část je zaměřena na samotnou stavbu jazykového modelu, opět je popsáno použití v systémech na TUL i v toolkitu Kaldi. Vytvářené modely jsou n-gramové.

Základem každého jazykového modelu je velké množství textových dat. Jejich kvalita značně ovlivňuje výsledky rozpoznávání. Pro velké diktovací programy je vhodné použít více zdrojů, velmi často se pro obecné modely používají texty z informačních serverů. Pokud je známá oblast výzkumu cílové aplikace, je vhodné jí přizpůsobit i jazykový model. Pro diktovací systém pro medicínu je tak vhodné použít data z tohoto prostředí. Velkým problémem může být mluvený charakter řeči, konkrétně nespisovnost a různá nářečí. Získat takováto kvalitní textová data pro jazykový model je velmi obtížné. Pro tuto práci byly použity texty z velkých amerických informačních serverů zaměřených na různé oblasti života.

2.4.1 Textová data

Většina rozsáhlých a kvalitních textových korpusů pro angličtinu je placená a nebyla tak pro tuto práci k dispozici. Místo toho byly využity informační servery, z nichž byly postahovány jednotlivé články a použity na stavbu výsledných

jazykových modelů. Pro vytvoření kvalitního modelu je potřeba až několik gigabytů jazykově bohatých dat. Jejich sběr bylo tedy potřeba zautomatizovat. Z tohoto důvodu byl navržen a implementován webový pavouk v jazyce Java, který data sbírá a ukládá do souborů. Jeho základem je HTML parser, který se stará o zisk potřebných dat z HTML dokumentu. Cílem nebylo vytvořit univerzálního pavouka, ale pavouky zaměřené na konkrétní stránky s důrazem na kontrolu stahovaného obsahu. Obrázek 4 zobrazuje základní strukturu parseru.



Obrázek 4: Diagram tříd – parser

Pod rozhraním a abstraktní třídou jsou parsery pro konkrétní stránky. Aplikace za běhu sbírá URL odkazy pouze na články na daném serveru a následně z nich ukládá jádro zprávy do souboru. Pro základní práci s DOM elementy byla použita knihovna jsoup [14]. Před spuštěním lze vybrat, z jaké oblasti chce uživatel články stáhnout a také z jaké doby.

Data byla stahováno ze dvou velkých amerických informačních serverů, Reuters [23] a The Guardian [27]. Oba servery nabízejí dostatek rozmanitých zpráv v americké angličtině z různých oblastí novinářské činnosti. Celkově bylo takto postahováno necelých 3,5 GB dat různých témat od počátku současného tisíciletí až do konce března roku 2014. Mezi tematické celky patří například světové zprávy, lokální zprávy pro Ameriku a Evropu, sport, technologie, finance a životní styl. Tabulka 6 zobrazuje množství stažených dat z jednotlivých informačních serverů.

Tabulka 6: Detail získaných dat pro jazykový model

web	velikost dat [GB]	rozsah let
Reuters	1,8	2006–2014
The Guardian	1,56	2000–2014

Takto získaná data ještě není vhodné použít k tvorbě jazykového modelu, je potřeba je předzpracovat. Při tomto procesu je text pročištěn a sjednocen, aby byl vhodný pro tvorbu modelů. Konkrétní úkony, ať už zaměřené na specifický jazyk nebo obecné, provedené s texty jsou popsány v následující kapitole.

2.4.2 Předzpracování textů

Při předzpracování textů je na stažená data aplikována sada skriptů, která je připraví do ideální podoby pro trénování finálních modelů. Tyto skripty je možné rozdělit do dvou základních kategorií:

- závislé na jazyce,
- nezávislé na jazyce.

Pro tuto práci byly využity jako základ skripty vytvořené pro předzpracování polštiny na Technické Univerzitě v Liberci. Skripty, obzvláště jazykově závislá část, byly silně upraveny. Celou operaci předzpracování lze rozdělit do několika bloků zaměřených na různé úpravy textu:

- hlavní skript,
- převod speciálních znaků do ASCII,
- odstranění HTML a XML tagů,
- **základní textové úpravy,**
- oddělení slov,
- **přepis dat**,
- **přepis čísel**,
- **kolokace**, výstupní soubory.

Tučně zobrazené jsou části ovlivněné konkrétním jazykem. Jednotlivé skripty jsou aplikovány v uvedeném pořadí. Výstup jednoho skriptu je vstupem toho dalšího. Veškeré skripty jsou napsané ve skriptovacím jazyce Perl. Perl je kvalitním

nástrojem pro práci s regulárními výrazy, které jsou velmi často používány při zpracování textu, při jeho nahrazování. Vstupem pro skripty je soubor obsahující cesty k textům, které mají být zpracovány. Tyto texty lze považovat za jeden korpus. Výstup jednotlivých částí předzpracování textu je ukázán na následujícím bloku textu.

```
She's the one that gave me $20. <br />
<p>Today's date is 1/4/2014.</p>
When you play the game of thrones, you win or you die...
```

V první části jsou převáděny speciální znaky na znaky standardnější. Tato část je nezávislá na jazyce. V uvedeném případě došlo k jediné záměně, 3 tečky byly převedeny na znak značící apoziopezi.

```
She's the one that gave me $20. <br />
<p>Today's date is 1/4/2014.</p>
When you play the game of thrones, you win or you die…
```

Textová data získaná pro tuto práci byla již při stahování zbavena HTML tagů, ale vytvořené skripty mají část zaměřenou i na tuto problematiku. Opět se jedná o část skriptů, která je nezávislá na daném jazyce. Výstup na uvedeném příkladu vypadá následovně.

```
She's the one that gave me $20.
Today's date is 1/4/2014.
When you play the game of thrones, you win or you die…
```

V následujícím kroku už jsou provedeny úpravy specifické pouze pro angličtinu. Cílem tohoto bodu je například sjednotit zkratky titulů, jednotek, měn a často používaných spojení s jejich rozvedenou formou [1]. Zároveň jsou zavedeny slovní třídy pro desetinnou čárku (v angličtině tečka) a procenta. Poslední důležitou částí je nadefinování a sjednocení základních formátů čísel, které jsou používány dalšími skripty.

Ukázka některých regulárních výrazů pro řešení konkrétních problémů následuje. V prvním řádku je symbol dolaru nahrazen za třídu dollar#, se kterou se dále pracuje. Ve druhém řádku dochází k prohození čísla a třídy dollar#, v mluvené řeči v angličtině je jejich pořadí opačné než v psané podobě. Poslední řádek ukazuje sjednocení tvarů zkratky s a bez tečky.

```
$sub_counter += $line=~s/\$/dollar#/g;
$sub_counter += $line=~s/(dollar#)(\d+)/$2 $1/gi;
$sub_counter += $line=~s/mrs\.{0,1}\s/mrs /gi;
```

Po vykonání tohoto skriptu má text následující podobu. Jednotka dolarů byla převedena na slovní třídu a umístěna na správné místo, aby odpovídala mluvené podobě jazyka. Zároveň bylo předpřipraveno datum pro další zpracování. Poslední věta zůstala beze změn.

```
She's the one that gave me 20 dollar#.
Today's date is 1 / 4 / 2014 .
When you play the game of thrones, you win or you die…
```

Další část slouží k odsazení interpunkčních znamének. Tento krok se provádí, aby slova ukončená interpunkcí nebyla při tvorbě jazykového modelu považována za slova nová. Tato část je z velké části jazykově nezávislá, ale je možné nalézt i výjimky. Jednou z nich může být například apostrof, který je v angličtině běžnou součástí slov a nesmí tak být odsazen. Výstup je následující.

```
She's the one that gave me 20 dollar# .
Today's date is 1 / 4 / 2014 .
When you play the game of thrones , you win or you die …
```

Navazující část je zaměřená na sjednocení formátu dat. V tomto kroku bylo důležité mít textová data z americké angličtiny, aby nedocházelo k omylům. Některé formáty dat pro britskou a americkou angličtinu se totiž liší, například americký formát mm/dd/yyyy má v britské angličtině tvar dd/mm/yyyy. Bylo nadefinováno několik základních formátů používaných právě v americké angličtině a ty byly následně sjednoceny do jednoho, odpovídajícího mluvené podobě. Přepisované formáty, uvedené na datu 11. 2. 1990, jsou následující:

- 11(th) February 1990,
- February 11(th) 1990,
- 02/11/1990,
- 1990-02-11.

Finální tvar vypadá následovně:

```
11#r of February 19# 90#
```

Zároveň jsou v tomto kroku přepisovány intervaly let a slova označující desetiletí (sixties označující šedesátá léta atd.). Výsledný text po aplikování všech pravidel vypadá následovně. Přípona #r u dne značí řadovou číslovku, je vloženo slovo of, které se jenom vyslovuje a rok je rozložen podle mluvené podoby.

```
she's the one that gave me 20 dollar# .
today's date is the 4#r of january 2# thousand and 14# .
when you play the game of thrones , you win or you die …
```

Předposlední částí je skript zaměřený na sjednocení čísel. Řeší například rozepisování čísel vyšších řádů (oddělovány čárkami po 3 cifrách), desetinných čísel a hodnot měn. Všechna čísla jsou přepsána pomocí slovních tříd do podoby odpovídající té mluvené. Číslovka 20 u dolarů je tak převedena na odpovídající slovní třídu.

```
she's the one that gave me 20# dollar# .
today's date is the 4#r of january 2# thousand and 14# .
when you play the game of thrones , you win or you die …
```

Závěrečný skript se stará o vygenerování souborů použitelných skripty pro tvorbu jazykového modelu na TUL. Jedná se o tři soubory pro každý textový soubor ve formátech `jzspoken000`, `jzspoken001` a `jzspoken002`. První z nich vypadá následovně, interpunkční znaménka jsou převedena na slovní reprezentaci vhodnou pro diktování.

```
she's the one that gave me 20# dollar# dot
today's date is the 4#r of january 2# thousand and 14# dot
when you play the game of thrones comma you win or you die
ellipsis
```

Druhý z nich interpunkční znaménka úplně odstraňuje a poslední nahrazuje všechny jedním symbolem. Tyto soubory jsou následně využívány při trénování jazykových modelů.

2.4.3 Trénování jazykových modelů

Podobně jako kapitola o trénování akustických modelů je i tato kapitola rozdělena na dvě základní části, první věnovanou trénování jazykových modelů na

Technické Univerzitě v Liberci a druhou, která se zabývá jazykovými modely v toolkitu Kaldi.

Trénování na TUL

Jazykové modely používané při rozpoznávání na Technické Univerzitě v Liberci jsou bigramové (kapitola 1.2.1). Vyšší řády nejsou podporovány. Trénovací skripty bylo potřeba opět upravit pro podporu americké angličtiny a vhodně nachystat vstupní soubory.

Textové vstupy pro tvorbu samotného jazykového modelu musí být ve formátech `jzspoken` vytvořených při předzpracování. Skript umožňuje počítání jazykového modelu na více korpusech, jednotlivé korpusy mohou mít různou váhu. Dalším důležitým vstupem je slovník ve formátu `lex` (kapitola 2.3.2), opět je možné kombinovat více slovníků. Posledním vstupem jsou speciální slovníky. Ty je nutné ručně vyrobit při předzpracování. Jedná se například o fonetické výslovnosti ke slovním třídám a různým zkratkám, aby mohly být správně napočítány bigramy. Slovník je opět ve formátu `lex`, následující ukázka je z dodatkového slovníku měn.

```
<item ft='dollar' p='dolE' m='dollar#' />
<item ft='dollar' p='dÁlE' m='dollar#' />
```

Před prvním sestavením jazykového modelu je také nutné připravit soubor `noise.lex.addon`, který v XML v kořenovém tagu `nanolexicon` uchovává, které znaky reprezentují ticho a hluky.

```
<item ft='[n::silence]' p='-' m='!noise' f='1'/>
<item ft='[h::stroke]' p='0' m='!noise' f='1'/>
```

Cesty k umístění slovníků a textových korpusů jsou uloženy v dalším XML souboru, se kterým už pracují samotné trénovací skripty. V konfiguračním souboru XML je možné vybrat, které slovníky a textové korpusy a s jakou váhou budou použity. Také je zde možné vybrat, které kroky trénování budou provedeny.

Skript byl upraven, aby dokázal pracovat s anglickými daty. Konvence pro konkrétní jazyky tak byly rozšířeny právě o podporu americké angličtiny. Samotný skript se skládá z několika relativně samostatných částí:

- načtení konfiguračního souboru,
- vytvoření slovníku,
- výpočet slovních párů pro korpusy,
- výpočet jazykového modelu (ARPA formát [2]),
- mixování jazykových modelů,
- konverze pro Newton Dictate.

Na začátku jsou načtena data z konfiguračního souboru představeného v této kapitole. Ze všech slovníků, tedy normálních i speciálních, je následně vytvořen veliký slovník. Slova z něj jsou použita při výpočtu slovních párů. Ty jsou počítány pomocí toolkitu SRILM (kapitola 2.2.3). Výsledný jazykový model je ve formátu ARPA. Následuje případné mixování jazykových modelů a finální konverze. Sekvence těchto kroků vede k vytvoření výsledného jazykového modelu ve formátu `ses`. Ten je použit při samotném rozpoznávání pomocí libereckého rozpoznávače.

Trénování pomocí Kaldi

Kaldi nemá samo o sobě žádné nástroje pro tvorbu jazykových modelů. V tomto ohledu se spoléhá na ostatní toolkity. Rozpoznávací skripty pracují s formátem ARPA. Pro vytvoření n-gramových modelů v tomto formátu je možné použít například toolkity SRILM [25], MITLM [20] nebo IRSTLM [12]. Pro tuto práci byly použity jazykové modely připravené pro liberecký rozpoznávač.

3 Vybrané experimenty

Důležitým bodem zadání bylo experimentální nalezení nejlepších akustických a jazykových modelů pro rozpoznávání spojité angličtiny rozpoznávačem vyvíjeným na Technické Univerzitě v Liberci. Tato kapitola ukazuje provedené experimenty a jejich výsledky. Vybrané experimenty jsou ještě srovnávány s rozpoznávačem v toolkitu Kaldi. Modely pro něj byly natrénována tak, aby co nejvíce odpovídaly modelům vytvořeným pro rozpoznávač vyvíjený na TUL. Úplné shody ale nebylo možné dosáhnout už jen z důvodu, že Kaldi je založené na crossword trifonech a rozpoznávač vyvíjený v Liberci pracuje s inter-word trifony. Pro tvorbu akustických modelů byly použity již dříve představené korpusy TIMIT a VoxForge. Tabulka 7 ukazuje přibližnou celkovou délku nahrávek jednotlivých korpusů.

Tabulka 7: Rozložení akustických dat

korpus	trénovací data [hod]	testovací data [hod]
TIMIT	4,1	1,3
VoxForge	69,3	2,8

Pro tvorbu různých jazykových modelů byla využita data představená v kapitole 2.4.1. Pro trénování jazykových modelů pro vybrané pokusy byly použity textové přepisy trénovacích dat jednotlivých korpusů. Tabulka 8 zobrazuje přehled jednotlivých textových korpusů a množství textu, které obsahují.

Tabulka 8: Rozložení jazykových dat

korpus	trénovací data [MB]
Reuters	1 800
The Guardian	1 560
TIMIT	0,22
VoxForge	2,42

Tabulka 4 představuje použité slovníky pro všechny experimenty. Hlavním cílem bylo vytvořit modely pro diktovací systémy pracující s velkými slovníky, proto byl menší slovník TIMIT použit jen v úvodních experimentech a následně byl

nahrazen složeným slovníkem. V následujících podkapitolách jsou představeny vybrané provedené experimenty a jejich výsledky jsou patřičně okomentovány.

3.1 Základní experiment

V době tohoto experimentu byly dostupné fonetické přepisy pouze pro korpus TIMIT bez anotací hluků. Veškeré tyto přepisy byly převedeny podle konverze představené v kapitole 2.3.6 a na takto připravených datech byl natrénován malý akustický model.

Akustický model:

- data: TIMIT, po konverzi bez anotovaných hluků,
- metoda: HMM, svázané trifony, 32 mixtur, 8 iterací.

Jazykový model:

- data: TIMIT, přepisy trénovacích dat,
- metoda: bigramový model.

Jako slovník byl pro tento pokus použit základní slovník dodávaný s korpusem TIMIT. Jazykový model byl ten nejjednodušší, založený jen na přepisech trénovacích dat. Experiment byl proveden na testovací podmnožině korpusu TIMIT. Pro porovnání výsledků byl přepsán základní recept pro TIMIT z toolkitu Kaldi z fonémového rozpoznávání na rozpoznávání slovní.

Tabulka 9: Srovnání rozpoznávačů na základním experimentu

Rozpoznávač	Accuracy [%]	Correctnes [%]
TUL	38,99	47,17
Kaldi	40,18	45,97

Relativní podobnost výsledků (viz Tabulka 9) posloužila jako ověření správnosti vytvořených trénovacích postupů a takto vzniklý akustický model byl použit pro pevné zarovnání pro zisk přepisů s anotovanými hluky.

3.2 Experimenty s anotovanými hluky

Anotace hluků do fonetických přepisů byla provedena pro zlepšení výsledků rozpoznávání. Z tohoto důvodu bylo prvním experimentem porovnání výsledků z minulého testu s výsledky získanými experimentem na datech s anotovanými hluky. Parametry trénování tedy také odpovídají předešlé úloze.

Akustický model:

- data: TIMIT, anotované hluky,
- metoda: HMM, svázané trifony, 32 mixtur, 8 iterací.

Jazykový model:

- data: TIMIT, přepisy trénovacích dat,
- metoda: bigramový model.

Slovníkem zůstal původní slovník z korpusu TIMIT a testování bylo provedeno na stejné podmnožině jako v předchozí úloze. Tabulka 10 zachycuje výsledky experimentu.

Tabulka 10: Vliv anotovaných hluků na rozpoznávání – TUL

TUL	Accuracy [%]	Correctnes [%]
bez hluků	38,99	47,17
anotované hluky	42,10	49,16

Z hlediska vyhodnocování kvality rozpoznávání řeči je více vypovídající veličina *Accuracy*, protože počítá i s chybami způsobenými chybným vložením slov do rozpoznané věty. Z tohoto důvodu se zavedení hluků do fonetických přepisů, jak je z výsledků patrné, pozitivně projevilo na kvalitě rozpoznávání. Některé typické rušivé elementy jsou nyní detekovány a neovlivňují tak výsledný rozpoznaný přepis věty.

Stejný experiment byl proveden i s toolkitem Kaldi. Tabulka 11 zobrazuje výsledky tohoto experimentu.

Tabulka 11: Vliv anotovaných hluků na rozpoznávání – Kaldi

Kaldi	Accuracy [%]	Correctnes [%]
bez hluků	40,18	45,97
anotované hluky	43,38	46,37

I v tomto případě se anotace hluků kladně projevila na kvalitě rozpoznávání. Pro všechny další experimenty byla proto používána data s anotovanými hluky.

3.3 Experimenty s akustickými daty

Veškeré dosud provedené experimenty byly založeny na malém akustickém modelu TIMIT. Malé množství akustických dat se ale negativně projevuje na kvalitě rozpoznávání, protože jednotlivé fonémy nejsou natrénovány pomocí dostatečného množství záznamů. Dalším krokem tedy bylo rozšíření zdrojových dat akustického modelu. Pro tento účel byly použity nahrávky z VoxForge. Fonetické přepisy byly získány při pevném zarovnání, kdy byly také anotovány hluky. Jako data pro jazykový model zůstaly stále jen přepisy trénovacích nahrávek korpusu TIMIT. Základní parametry pro experimenty byly následující.

Akustický model:

- data: TIMIT + VoxForge, anotované hluky,
- metoda: HMM, svázané trifony, 32 mixtur, 8 iterací.

Jazykový model:

- data: TIMIT, přepisy trénovacích dat,
- metoda: bigramový model.

Použitý slovník byl stále původní TIMIT slovník. Modely byly otestovány pomocí testovací podmnožiny dat korpusu TIMIT. Experiment byl opět nejprve proveden pomocí univerzitního rozpoznávače, viz Tabulka 12.

Tabulka 12: Experiment s rozšířeným akustickým modelem – TUL

Akustická data	Accuracy [%]	Correctnes [%]
TIMIT	42,10	49,16
TIMIT + VoxForge	43,09	51,40

Tabulka 13 obsahuje výsledky experimentu provedené pomocí Kaldi.

Tabulka 13: Experiment s rozšířeným akustickým modelem – Kaldi

Akustická data	Accuracy [%]	Correctnes [%]
TIMIT	43,38	46,37
TIMIT + VoxForge	44,60	48,96

Zvýšením množství zdrojových dat akustického modelu došlo k mírnému zlepšení úspěšnosti rozpoznávání u obou použitých rozpoznávačů. Pokrok není u testovací sady TIMIT nikterak oslňující, ale modely se staly robustnějšími a u jiných testovacích sad by mohl být rozdíl mnohem markantnější. Z provedeného experimentu je i tak patrné, že zvětšení trénovací sady se kvalitativně projevilo na výsledcích. Pro češtinu jsou k dispozici na TUL až stovky hodin trénovacích dat získaných během vývoje systému. Pro účely této práce bylo množství akustických dat značně omezeno. Zvětšení trénovací množiny by se dále projevilo zlepšováním dosažených výsledků.

3.4 Experimenty s jazykovými daty

Rozpoznávač Kaldi umožňuje použití n-gramových jazykových modelů vyšších řádů. Z tohoto důvodu byl proveden experiment zkoumající vliv trigramového modelu na úspěšnost rozpoznávání oproti bigramovému modelu využívanému u všech ostatních experimentů.

Akustický model:

- data: TIMIT, anotované hluky,
- metoda: HMM, svázané trifony, 32 mixtur, 8 iterací.

Jazykový model:

- data: TIMIT, přepisy trénovacích dat,
- metoda: bigramový, trigramový model.

Experiment byl proveden na testovací sadě TIMIT s využitím stejnojmenného slovníku. Tabulka 14 obsahuje výsledky experimentu.

Tabulka 14: Experiment s bigramovým a trigramovým modelem

n-gram	Accuracy [%]	Correctnes [%]
bigram	43,38	46,37
trigram	44,77	47,07

Použití trigramového jazykového modelu se pozitivně projevilo na úspěšnosti rozpoznávání experimentu založeného na malém slovníku. Trigramové modely však bohužel nebylo možné použít pro experimenty s rozšířeným slovníkem z důvodu vysoké paměťové a výpočetní náročnosti. Aplikace vyšších řádů by však byla přínosná.

Veškeré dosud provedené experimenty byly prováděny s jednoduchým jazykovým modelem založeným jen na přepisech trénovacích dat korpusu TIMIT. Dalším logickým krokem po zvýšení množství zdrojových dat akustického modelu bylo použití natrénovaných jazykových modelů na textech ze zpravodajských serverů. Prvotní experiment byl proveden s jazykovým modelem založeným na textech z informačního serveru Reuters.

Akustický model:

- data: TIMIT + VoxForge, anotované hluky,
- metoda: HMM, svázané trifony, 32 mixtur, 8 iterací.

Jazykový model:

- data: Reuters,
- metoda: bigramový model.

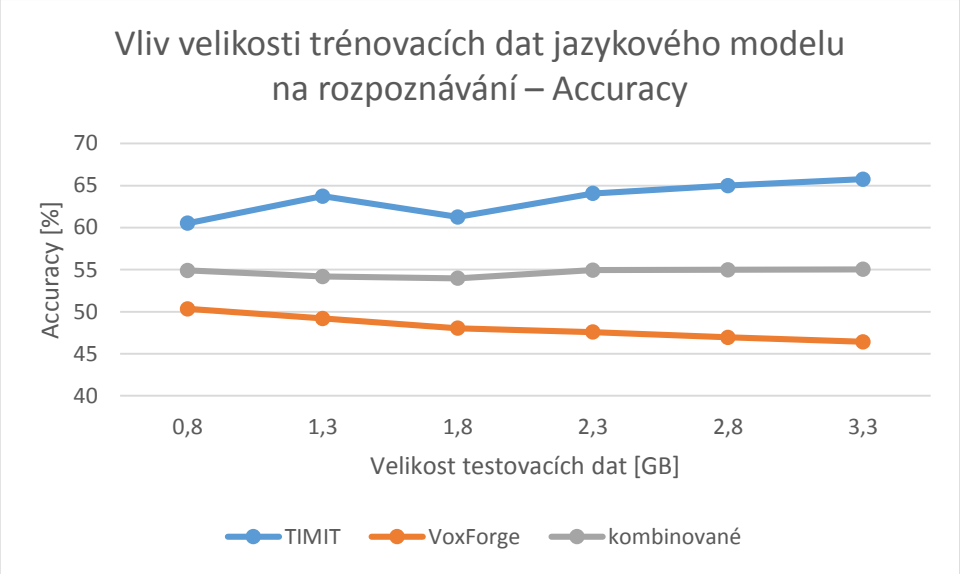
Pro tento experiment už byl použit složený slovník a také testovací sada vybraná z korpusu VoxForge. Experimenty byly provedeny pomocí rozpoznávače vyvíjeného na TUL. Tabulka 15 zobrazuje výsledky experimentu.

Tabulka 15: Výsledky experimentů s jazykovým modelem Reuters

Testovací data	Accuracy [%]	Correctnes [%]
TIMIT	61,27	65,73
VoxForge	48,04	62,89
kombinace	53,98	64,22

Na získaných výsledcích lze vidět další zlepšení v úspěšnosti rozpoznávání, u testovací sady TIMIT se kvalita rozpoznávání zlepšila téměř o 20 %. Zároveň výsledky ukazují, že kvalita rozpoznávání také úzce souvisí se samotnými testovacími daty. Zatímco data z korpusu TIMIT jsou studiově nahraná a speciálně připravená pro účely rozpoznávání, nahrávky z VoxForge jsou kompilací dat různé kvality od internetové komunity a úspěšnost rozpoznávání je na nich tak nižší.

Důležitým faktorem ovlivňujícím výsledky rozpoznávání je také velikost a tematické zaměření samotného modelu. Následující sada experimentů byla provedena na stejném akustickém modelu, slovníku a testovacích datech jako předešlý představený pokus. Experimenty se od sebe lišily jen velikostí použitého jazykového modelu. Graf 2 zobrazuje výsledky provedeného experimentu pro veličinu *Accuracy*.

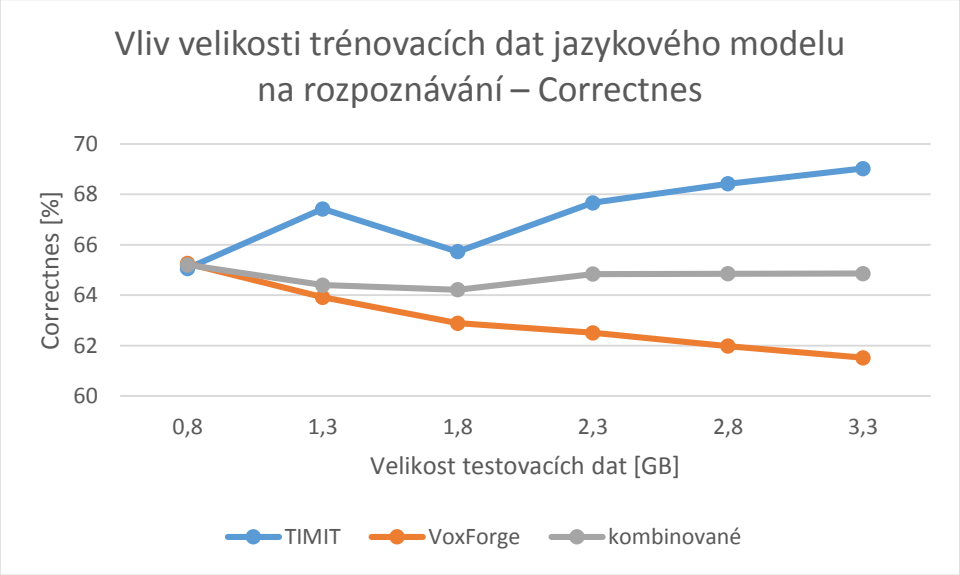


Graf 2: Vliv velikosti trénovacích dat jazykového modelu na Accuracy

U testovacích množin TIMIT a kombinované se zvětšování trénovacích jazykových dat projevilo pozitivně na úspěšnosti rozpoznávání. U množiny VoxForge naopak došlo ke zhoršení výsledků. To je způsobeno tematickým zaměřením jazykového modelu a testovacích dat. Zatímco jazykový model je založený na datech ze zpravodajských serverů, testovací data VoxForge obsahují spontánní promluvy od komunity z různých životních oblastí. Testovací sada TIMIT je založená na jazykově bohatých spisovných větách americké angličtiny. Pozitivní

dopad velikosti jazykového modelu by se nejvíce projevil na testovacích datech ze zpravodajské oblasti.

Graf 3 ukazuje výsledky stejného experimentu pro veličinu *Correctnes*.



Graf 3: Vliv velikosti trénovacích dat jazykového modelu na Correctnes

Výsledky pro veličinu *Correctnes* mají stejnou tendenci jako u *Accuracy*. Opět jsou výsledky ovlivněny zaměřením trénovacích dat jazykových modelů a testovací sadou. Nejlepších výsledků, podobně jako u *Accuracy*, by bylo dosaženo u testovací sady zaměřené na zpravodajství.

3.5 Závěrečné porovnání

Závěrečným experimentem bylo srovnání rozpoznávače vyvíjeného na Technické Univerzitě v Liberci s toolkitem Kaldi za použití nejlepších akustických a jazykových modelů. Základní charakteristika experimentu byla tedy následující:

Akustický model:

- data: TIMIT + VoxForge, anotované hluky,
- metoda: HMM, svázané trifony, 32 mixtur, 8 iterací.

Jazykový model:

- data: Reuters + The Guardian,
- metoda: bigramový model.

Slovník byl opět použit složený a jako testovací data byla použita kombinovaná podmnožina korpusů TIMIT a VoxForge. Tabulka 16 ukazuje výsledky závěrečného experimentu.

Tabulka 16: Srovnání rozpoznávačů na závěrečném experimentu

Rozpoznávač	Accuracy [%]	Correctnes [%]
TUL	55,04	64,86
Kaldi	57,11	63,75

Úspěšnost rozpoznávání je u rozpoznávače vyvíjeného na Technické Univerzitě v Liberci a rozpoznávače přítomného v Kaldi velmi podobná. Kaldi dosahuje lepších výsledků u *Accuracy*, zatímco univerzitní rozpoznávač má lepší výsledky u veličiny *Correctnes*. Liberecký rozpoznávač je také založený na inter-word trifonech, zatímco rozpoznávač v Kaldi využívá crossword trifonů, které ale nejsou tak vhodné pro online rozpoznávání. Navíc Kaldi používá při trénování přiřazení jednotlivých promluv k mluvčím. Většina ostatních parametrů byla zvolena tak, aby experimenty byly co nejobjektivnější. Experimentálně získané hodnoty z posledního experimentu jsou srovnatelné a veškeré provedené úpravy, které byly součástí této práce, se kladně projevily na úspěšnosti rozpoznávání.

Závěr

V rámci diplomové práce byla shromážděna akustická a jazyková data pro anglický jazyk. Tato data byla na základě vytvořených skriptů předzpracována a připravena pro použití při trénování a testování. Pro tvorbu modelů založených na angličtině byly navrženy trénovací skripty, které umožňují jejich snadné vytváření. Akustické i jazykové modely byly následně experimentálně otestovány a nejlepší z nich byly převedeny do formátu použitelného v aplikaci Newton Dictate, kde by měly sloužit jako ukázka schopností rozpoznávače vyvíjeného na Technické Univerzitě v Liberci.

Pro popis zvukové stránky anglického jazyka byla vybrána fonetická abeceda skládající se z 39 fonémů. Na jejím základě byl sestaven slovník pokrývající velkou část americké angličtiny vycházející z volně dostupného slovníku CMU. Zdrojem akustických dat se stal komerční korpus TIMIT obsahující několik hodin nahrávek mluvené americké angličtiny různých nářečí s ručně vytvořenými fonetickými přepisy. Pro vytvoření robustnějších modelů byla tato data rozšířena o nahrávky z komunitního projektu VoxForge. Shromážděná data byla následně pomocí skriptů parametrizována a podmnožina TIMIT byla použita k natrénování prvotního akustického modelu toolkitem HTK. Tento model byl použit v úloze pevného zarovnání pro zisk fonetických přepisů pro nahrávky z projektu VoxForge a pro anotaci ticha a různých hluků u všech nahrávek. Výsledky pevného zarovnání také posloužily k strojovému vyfiltrování nekvalitních akustických záznamů, které by se negativně projevily na úspěšnosti rozpoznávání. Pro angličtinu byly navrženy trénovací skripty pomocí toolkitů HTK a Kaldi. Veškeré vytvářené modely pomocí těchto toolkitů byly fonémové skryté markovské modely.

Pro tvorbu jazykových modelů nebyl k dispozici žádný textový korpus. Z tohoto důvodu byl navržen webový pavouk, jehož cílem byl sběr textových dat z vybraných zpravodajských serverů. Konkrétně byly implementovány dvě třídy pod základním rozhraním stahující data z amerických serverů Reuters a The Guardian. Takto získané texty byly následně automaticky předzpracovány z hlediska jazykově závislého i nezávislého. Mezi závislé lze zařadit automatické sjednocování čísel, dat a zkratek do jednotného formátu. Zde bylo využito jedné z hlavních předností

skriptovacího jazyka Perl, konkrétně práce s regulárními výrazy. Mezi jazykově nezávislé úkony patřilo například odsazení interpunkčních znamének od slov, aby nedocházelo k chybným výpočtům při sestavování jazykového modelu. Výsledné trénovací skripty vytváří bigramové jazykové modely použitelné jak v rozpoznávači vyvíjeném na Technické Univerzitě v Liberci, tak v Kaldi.

Veškeré experimenty byly prováděny pomocí univerzitního rozpoznávače. Některé významné experimenty byly pro srovnání také uskutečněny pomocí rozpoznávače dostupného v toolkitu Kaldi. Jednotlivé pokusy odrážejí prováděné úpravy na akustickém i jazykovém modelu a jejich výsledky ilustrují vliv těchto úprav na úspěšnost rozpoznávání. Největší podíl na zvýšení rozpoznávání měla anotace ticha a hluků a zavedení kvalitního jazykového modelu. Během práce se podařilo zlepšit úspěšnost rozpoznávání až o 25 % na stejné testovací sadě s mnohonásobně zvětšeným slovníkem.

Výsledky rozpoznávání jsou ale stále limitovány množstvím akustických dat, modely vytvářené pro češtinu na Technické Univerzitě v Liberci jsou založené až na stovkách hodin různých nahrávek. Pro angličtinu i po přidání záznamů z korpusu VoxForge bylo k dispozici jen několik desítek hodin. S rostoucí velikostí akustických dat by se dále zvyšovala úspěšnost a robustnost rozpoznávače. Tematické zaměření a velikost jazykového modelu jsou také výrazným faktorem ovlivňujícím výsledky rozpoznávání. Vytvářené modely byly navrženy univerzálně pro diktovací aplikaci. To ale může být nevýhodou u aplikací zaměřených na specifickou problematiku. Úspěšnost rozpoznávání v neposlední řadě také ovlivňuje samotná testovací sada a použitý slovník. Slova obsažená v testovacích datech, ale chybějící ve slovníku nemají šanci být nikdy správně rozpoznána.

Hlavní přínosy práce:

- skripty pro předzpracování dat pro angličtinu,
- skripty pro trénování akustických a jazykových modelů pro angličtinu pro rozpoznávač vyvíjený na Technické Univerzitě v Liberci a pro Kaldi,
- použití vytvořených modelů pro zisk fonetických přepisů dalších dat,
- experimentální ověření úspěšnosti rozpoznávače vyvíjeného na TUL,
- podpora americké angličtiny pro rozpoznávač TUL.

Na diplomovou práci by bylo možné dále navázat. Rozšiřování zdrojových akustických, lexikálních i jazykových dat by vedlo k dosažení ještě lepších výsledků úspěšnosti rozpoznávání. V úvahu také připadají alternativní méně používané metody vytváření akustických i jazykových modelů a následné porovnávání výsledků experimentů s výsledky prezentovanými v této práci.

Seznam použité literatury

[1] Abbreviations. *Capital Community College* [online]. [2014] [cit. 2014-04-14]. Dostupné z: http://grammar.ccc.commnet.edu/grammar/abbreviations.htm

[2] ARPA Format for n-grams. *SRI International's STAR Laboratory* [online]. ©2011 [cit. 2014-04-16]. Dostupné z: http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html

[3] *Free Speech... Recognition: voxforge.org* [online]. © 2006-2014 [cit. 2014-03-17]. Dostupné z: http://www.voxforge.org/home

[4] The GNU General Public License v3.0: GNU Project – Free Software Foundation. *The GNU Operating System* [online]. © 1996-, 14.3.2014 [cit. 2014-03-17]. Dostupné z: https://www.gnu.org/copyleft/gpl.html

[5] GRUHN, Rainer E., Wolfgang MINKER a Satoshi NAKAMURA. *Statistical Pronunciation Modeling for Non-Native Speech Processing: Automatic Speech Recognition*. Berlin: Springer, 2011, s. 5-17. ISBN 978-3-642-19585-3.

[6] HALDAR, Rishin a MUKHOPADHYAY. *Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. Calcutta*, India, 2011. Dostupné z: http://arxiv.org/ftp/arxiv/papers/1101/1101.1232.pdf

[7] *HTK Speech Recognition Kit* [online]. [2009] [cit. 2014-03-31]. Dostupné z: http://htk.eng.cam.ac.uk/

[8] HUANG, Xuedong. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Vyd. 1. New Jersey: Prentice-Hall, 2001. ISBN 01-302-2616-5.

[9] CHEN, Stanley F. a Joshua GOODMAN. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language* [online]. 1998, č. 13, s. 359-394 [cit. 2014-05-02]. Dostupné z: http://u.cs.biu.ac.il/~yogo/courses/mt2013/papers/chen-goodman-99.pdf

[10] IPA: Alphabet. *IPA: International Phonetic Association* [online]. [1888] [cit. 2014-03-24]. Dostupné z: http://www.langsci.ucl.ac.uk/ipa/ipachart.html

[11] IPŠIĆ, Ivo. *Speech technologies*. Rijeka: InTech, 2011, s. 285-302. ISBN 9789533079967.

[12] IRSTLM (IRST Language Modelling Toolkit). *Human Language Technology* [online]. © 2013 [cit. 2014-04-16]. Dostupné z: https://hlt.fbk.eu/technologies/irstlm-irst-language-modelling-toolkit

[13] JURAFSKY, Dan a James H MARTIN. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd ed. Upper Saddle River: Pearson Education, 2008, 988 s. ISBN 978-0-13-187321-6.

[14] *Jsoup: Java HTML Parser* [online]. © 2009 – 2013 [cit. 2014-04-13]. Dostupné z: http://jsoup.org/

[15] *Kaldi* [online]. [2014] [cit. 2014-03-31]. Dostupné z: http://htk.eng.cam.ac.uk/

[16] LOGIOS Lexicon Tool. *CMU Lexicon Tool* [online]. [2008] [cit. 2014-03-31]. Dostupné z: http://www.speech.cs.cmu.edu/tools/lextool.html

[17] Laboratoř počítačového zpracování řeči. *Ústav Informačních technologií a elektroniky* (ITE) [online]. ©2014 [cit. 2014-03-24]. Dostupné z: https://www.ite.tul.cz/speechlab/index.php

[18] *Linguistic Data Consortium* [online]. © 1992-2014 [cit. 2014-03-17]. Dostupné z: https://www.ldc.upenn.edu/

[19] MANNING, Christopher D. *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press, c1999, 680 s. ISBN 02-621-3360-1.

[20] *MIT Language Modeling Toolkit* [online]. ©2009 [cit. 2014-04-16]. Dostupné z: https://code.google.com/p/mitlm/

[21] NOUZA, Jan. *On the Speech Feature Selection Problem: Are Dynamic Features more Important than the Static Ones?*. Proc. Of EUROSPEECH'95 Conference, Madrid, Spain, Sept. 1995, s. 919-923.

[22] ROCH, Marie. *Acoustic Modeling for Speech & Speaker Recognition: IPA/CMU/TIMIT Phone Mappings and American English Examples. In: Speech Processing* [online]. [2014] [cit. 2014-04-08]. Dostupné z: http://roch.sdsu.edu/cs682/IPA-CMU-TIMITPhoneset.pdf

[23] *Reuters* [online]. [2014] [cit. 2014-04-13]. Dostupné z: http://www.reuters.com/

[24] *Řeč a počítač: principy hlasové komunikace, úlohy, metody a aplikace : sborník článků*. Vyd. 1. Editor Jan Nouza, Zbyněk Koldovský, Robert Vích. Liberec: Technická univerzita v Liberci, 2009, 235 s. ISBN 978-80-7372-548-8.

[25] SRILM – The SRI Language Modeling Toolkit. *SRI International* [online]. 2013 [cit. 2014-03-31]. Dostupné z: http://www.speech.sri.com/projects/srilm/

[26] The CMU Pronouncing Dictionary. *Carnegie Mellon School of Computer Science* [online]. [2008] [cit. 2014-03-24]. Dostupné z: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[27] *The Guardian* [online]. © 2014 [cit. 2014-04-13]. Dostupné z: http://www.theguardian.com/

[28] YOUNG, Steve, Dan KERSHAW, Julian ODELL, Dave OLLASON, Valtcho VALTCHEV a Phil WOODLAND. MICROSOFT CORPORATION. *The HTK Book*. 3. Vyd. 2000, 271 s.

A Obsah přiloženého CD

Přiložené CD obsahuje kromě této diplomové zprávy nejdůležitější součásti praktické práce a je rozčleněno do adresářové struktury.

- Adresář dokumentace
  - obsahuje dokumentaci ve formátech `pdf` a `docx`.
  - obsahuje kopii zadání ve formátu `pdf`.
- Adresář modely
  - obsahuje akustický model ve formátu `channel`.
  - obsahuje jazykový model ve formátu `ses`.
- Adresář fonetická abeceda
  - obsahuje fonetickou abecedu pro angličtinu ve formátech `abc` a `htk`.
- Adresář skripty a programy
  - obsahuje podpůrné a trénovací skripty a programy pro HTK i Kaldi.
- Adresář slovníky
  - obsahuje hlavní a pomocné slovníky ve formátech `lex`, `HTK` a `kaldi.`
- Adresář fonetické přepisy
  - obsahuje fonetické přepisy dat získané pevným zarovnáním.
  - obsahuje inter-word `mlf.triphones` soubor.